

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2017

Bc. Matěj Klapal



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**TRASOVÁNÍ POHYBU OBJEKTŮ S POMOCÍ
POČÍTAČOVÉHO VIDĚNÍ**

OBJECT TRACKING USING COMPUTER VISION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Matěj Klapal

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Radim Burget, Ph.D.

BRNO 2017

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Matěj Klapal

ID: 158159

Ročník: 2

Akademický rok: 2016/17

NÁZEV TÉMATU:

Trasování pohybu objektů s pomocí počítačového vidění

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s problematikou detekce a trasování pohybu objektů před kamerou. Navrhněte algoritmus, který bude schopen sledovat počet pohybujících se objektů na různých příkladech videí a komentujte, ve kterých případech algoritmus selhává. Navrhněte vylepšení algoritmu. Dosažené výsledky vhodně zhodnoťte a prezentujte na vybraném příkladě, jak se chová oproti standardní verzi, dále také komentujte, na kterých příkladech algoritmus stále (či nově) selhává.

DOPORUČENÁ LITERATURA:

[1] Bradski, Gary, and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.", 2008.

[2] Challa, Sudha. Fundamentals of object tracking. Cambridge University Press, 2011.

Termín zadání: 1.2.2017

Termín odevzdání: 24.5.2017

Vedoucí práce: doc. Ing. Radim Burget, Ph.D.

Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato diplomová práce je věnována možnostem sledování pohybu objektů pomocí algoritmů počítačového vidění. Úvodní kapitoly obsahují přehled využívaných metod pro eliminaci pozadí, uvedeny jsou také základní přístupy k detekci. Práce dále zmiňuje i řadu algoritmů umožňujících trasování a predikci pohybu. Další část práce informuje o algoritmech implementovaných ve výsledném programu a jeho grafickém rozhraní. V závěru je uvedeno zhodnocení a porovnání původního a modifikovaného algoritmu.

KLÍČOVÁ SLOVA

Trasování objektů, počítačové vidění, OpenCV, eliminace pozadí, Mixture of Gaussians, Kalmanův filtr

ABSTRACT

This diploma thesis deals with possibilities of tracking object movement using computer vision algorithms. First chapters contain review of methods used for background subtraction, there are also listed basic detection approaches and thesis also mentions algorithms which allows tracking and movement prediction. Next part of this work informs about algorithms implemented in resulting software and its graphical user interface. Evaluation and comparison of original and modified algorithms is stationed at the end of this text.

KEYWORDS

Object tracking, computer vision, OpenCV, background subtraction, Mixture of Gaussians, Kalman filter

KLAPAL, Matěj *Trasování pohybu objektů s pomocí počítačového vidění*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 57 s. Vedoucí práce byl doc. Ing. Radim Burget, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Trasování pohybu objektů s pomocí počítačového vidění“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Ing. Radimu Burgetovi, Ph.D., za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....
podpis autora(-ky)

OBSAH

Úvod	11
1 Eliminace pozadí	12
1.1 Metoda Frame Difference	12
1.2 Metoda Approximate Median	12
1.3 Metoda Running Gaussian Average	13
1.4 Metoda Mixture of Gaussians (MOG)	14
1.5 Samoorganizační neuronové sítě	15
1.5.1 MIL-SOBS	16
2 Detekce a značení objektů	17
2.1 Region	17
2.2 Kontura	18
2.3 Příznaky	18
2.4 Model	18
2.4.1 Model postavy	19
2.4.2 Modely pohybu	20
3 Sledování a predikce pohybu	21
3.1 Kalmanův filtr	21
3.2 Optický tok	22
3.2.1 Horn-Schunck	23
3.2.2 Lucas-Kanade	24
4 Návrh a implementace	25
4.1 Knihovna OpenCV	25
4.2 Výsledná aplikace	26
4.2.1 Analýza pohybu	27
4.2.2 Trasování objektů	30
4.2.3 Eliminace stříhu	31
4.3 Grafické rozhraní	31
4.3.1 Nastavení	32
4.3.2 Základní	32
4.3.3 Pokročilé	34
4.3.4 Zóny	36
4.3.5 Hranice	38

5	Zhodnocení	39
5.1	Testovací sada	39
5.1.1	Kritéria hodnocení	40
5.1.2	Přítomné chyby	41
5.2	Výsledky	43
5.2.1	Původní algoritmus	43
5.2.2	Modifikovaný algoritmus	45
5.2.3	Porovnání	46
6	Závěr	50
	Literatura	51
	Seznam symbolů, veličin a zkratk	55
	Seznam příloh	56
A	Obsah přiloženého CD	57

SEZNAM OBRÁZKŮ

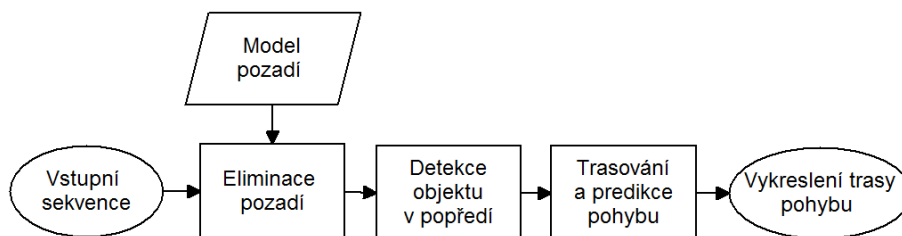
1	Postup trasování objektů	11
1.1	Gaussovo rozdělení pravděpodobnosti	14
1.2	Obrázek (vlevo) a příslušná neuronová síť (vpravo)	15
2.1	Detekce a blob	17
4.1	Blokový diagram eliminace pozadí	27
4.2	Originální snímek a výsledky morfologických operací	28
4.3	Výsledná maska popředí	29
4.4	Hlavní okno aplikace	32
4.5	Okno s binární maskou popředí	33
4.6	Okno nastavení – Základní	34
4.7	Okno nastavení – Pokročilé	35
4.8	Okno nastavení – Zóny	37
4.9	Okno nastavení – Hranice	38
5.1	Příklady testovacích sekvencí	40
5.2	Chyba predikce	42
5.3	Segmentovaná detekce	43
5.4	Sloučená detekce	43
5.5	Porovnání hodnot TPR	47
5.6	Porovnání hodnot PPV	47
5.7	Porovnání hodnot F_1	47
5.8	Opravené chyby (vlevo původní algoritmus, vpravo modifikace)	48

SEZNAM TABULEK

4.1	Příklad logu	33
5.1	Testovací sada	39
5.2	Výsledky původního algoritmu	44
5.3	Výsledky modifikovaného algoritmu	45
5.4	Porovnání výsledků	46

ÚVOD

Počítačové vidění je v současné době hojně rozvíjeným odvětvím. Základem je snaha, aby počítače prostřednictvím kamer a správného softwaru dokázaly pochopit danou scénu podobně jak ji chápe lidský mozek. Počítačové vidění nachází své uplatnění v oblastech zdravotnictví, pro diagnostické účely, rozpoznávání chování pacientů apod. Hojně je využíváno také v oblastech pracujících v reálném čase. Tato práce se zabývá sledováním pohybu objektů před statickou kamerou. Toto téma může nalézt uplatnění v zabezpečovací technice, kdy je sledován daný prostor a při nedovolené akci útočníka dojde například ke spuštění poplachu. Trasování a také počítání objektů může být užitečné v případě potřeby monitorovat pohyb například na dálnicích a vícepruhových silnicích. Trasování probíhá v několika krocích, které jsou uvedeny na zjednodušeném schématu na obrázku 1.



Obr. 1: Postup trasování objektů

Nejprve je získán model pozadí, který slouží k odstranění pozadí z dané sekvence. Následujícím krokem je takto získaná data na popředí správně zpracovat a označit je jako objekty. Následně je určena poloha těchto objektů a na základě předchozího pohybu je predikováno, jakým směrem se sledované objekty budou dále pohybovat.

Metodám umožňujícím eliminaci pozadí je věnována kapitola 1. Jsou zde uvedeny nejznámější a hojně využívané metody. Kapitola číslo 2 se zabývá možnostmi přístupu k objektům získaným předchozím zpracováním. Následující kapitola uvádí algoritmy použitelné za účelem predikce a trasování pohybu objektů. 4. kapitola je věnována samotnému programu. Jsou zde uvedeny algoritmy implementované ve výsledné aplikaci a je představeno grafické rozhraní. Jsou uvedeny jeho možnosti a je vysvětleno, jak s ním pracovat a jak se změna nastavení projeví na výsledné funkčnosti. V následující kapitole 5 jsou uvedeny výsledky a zhodnocena úspěšnost výsledného algoritmu, spolu s porovnáním s původní verzí, ze které program vychází.

1 ELIMINACE POZADÍ

Na úspěšnosti algoritmu pro trasování se velkou měrou podílí kvalita použité metody pro odstranění pozadí. Při návrhu je třeba brát v potaz přítomnost dynamicky se měnících vlastností obrazu. Jsou jimi ve venkovním prostředí například změna osvětlení v závislosti na denní době, větve stromů hýbající se pod vlivem větru či kašny nebo fontánky, které slouží k dekoračním účelům. Z těchto důvodů je vhodné vytvořit model pozadí, který se bude adaptivně měnit v závislosti na čase nebo na nastavených parametrech. V podkapitolách níže jsou uvedeny běžně používané metody. [1]

1.1 Metoda Frame Difference

Metoda Frame Difference je nejjednodušší metodou pro odstranění pozadí. Spočívá v odečtení aktuálního snímku od snímku specifikovaného jako snímek pozadí. Zpravidla je jako tento snímek použit první snímek z videa. Rozdíl mezi pixely dvou snímků (T_Δ) je pak porovnán s hodnotou prahu T_s , a když je $T_\Delta > T_s$ spadá pixel do popředí.

I přes nízkou výpočetní náročnost není tato metoda příliš vhodná pro použití v reálném čase, protože nedochází k žádným adaptivním změnám pozadí a nastavený model tak zůstává stejný po celou dobu zpracovávání. [2], [3].

1.2 Metoda Approximate Median

Tato metoda spočívá v ukládání několika předchozích snímků do paměti a snímek popisující pozadí je určen jejich mediánem. Medián je statisticky definován jako prostřední hodnota uspořádaného celku. V obraze může být tímto uspořádaným celkem například n -tice vzestupně seřazených odpovídajících pixelů. Mediánem je v tomto případě ta hodnota, která se nachází uprostřed této uspořádané řady. Snímek popředí je získán obdobně jako v předchozí metodě odečtením aktuálního snímku od pozadí. Souběžně s tím dochází k aproximaci mediánu vzájemným porovnáním těchto dvou snímků. Je-li hodnota jasu pixelu na pozadí menší než hodnota pixelu v aktuálním snímku, je hodnota jasu na pozadí inkrementována o jednu. Při opačné nerovnosti je pixel na pozadí naopak o jednu zmenšen.

Tato metoda je již vhodnější pro použití v reálném čase a při dynamických změnách pozadí, neboť zde již není porovnáván pouze jeden model pozadí, ale pozadí je naopak průběžně aktualizováno. Metoda je však poměrně paměťově náročná, z důvodu nutnosti ukládat do paměti celé snímky. [2], [3], [4], [5].

1.3 Metoda Running Gaussian Average

Metoda vychází z předpokladu, že hodnoty jasu pixelů mohou být modelovány pomocí Gaussova (normálního) rozdělení pravděpodobnosti, tedy, že hodnoty pixelů pozadí se s největší pravděpodobností pohybují kolem dané střední hodnoty. Metoda spočívá v získání pravděpodobnostní funkce pro hodnoty n -tice posledních snímků. Střední hodnota μ_t této funkce je určena s každým novým snímkem dle vztahu 1.1 a obdobně pro hodnotu rozptylu σ_t^2 dle vztahu 1.2.

$$\mu_t = \alpha \cdot I_t + (1 - \alpha) \cdot \mu_{t-1}, \quad (1.1)$$

$$\sigma_t^2 = \alpha \cdot d^2 + (1 - \alpha) \cdot \sigma_{t-1}^2, \quad (1.2)$$

$$d^2 = |(I_t - \mu_t)|, \quad (1.3)$$

kde

$\alpha \in (0, 1)$ je faktor adaptace, určuje jak rychle se bude μ_t a σ_t^2 měnit,

I_t je hodnota intenzity aktuálního pixelu,

d^2 je absolutní hodnota rozdílu mezi střední hodnotou a intenzitou aktuálního pixelu,

μ_{t-1} je střední hodnota a σ_{t-1}^2 je rozptyl předchozího snímku.

Jakmile jsou určeny parametry pro všechny pixely aktuálního snímku, dojde k jejich vzájemnému porovnání s příslušnými pixely dle nerovnice:

$$I_t - \mu_t \leq k \cdot \sigma_t, \quad (1.4)$$

kde k je hodnota volitelného prahu.

Platí-li vztah (1.4), pixel spadá do pozadí. V opačném případě se jedná o pixel v popředí.

Úspěšnost odstranění pozadí závisí na vhodně zvoleném prahu k . Úskalí nastává při volbě adaptačního faktoru α , je-li ve scéně přítomno více objektů, které se pohybují různou rychlostí pak dochází k nesprávnému odstranění pozadí. Toho by bylo možné se vyvarovat nastavováním parametrů k a α pro různé části scény jinak.

Výhodou oproti předchozí metodě je především úspora paměti, neboť již není třeba uchovávat v paměti celé snímky. Metoda si vystačí pouze s dvojicí parametrů (μ, σ^2) . Úspora paměťových prostředků je však bohužel vyvážena vyšší výpočetní náročností z důvodu nutnosti provádět výpočty pro každý pixel. I přes to je však metoda již vhodná pro použití v reálném čase.[2], [3].

1.4 Metoda Mixture of Gaussians (MOG)

Jak již název napovídá, tato metoda taktéž využívá Gaussovo rozdělení pravděpodobnosti. Pro všechny pixely každého možného povrchu je určena řada funkcí hustoty pravděpodobnosti. Pravděpodobnostní funkce je, pro případ jednokanálového obrazu ve stupních šedi, definována ve tvaru dle vztahu 1.5.

$$f(x) = \frac{1}{\sigma_n \sqrt{2\pi}} e^{-\frac{(x-\mu_n)^2}{2\sigma_n^2}}, \quad (1.5)$$

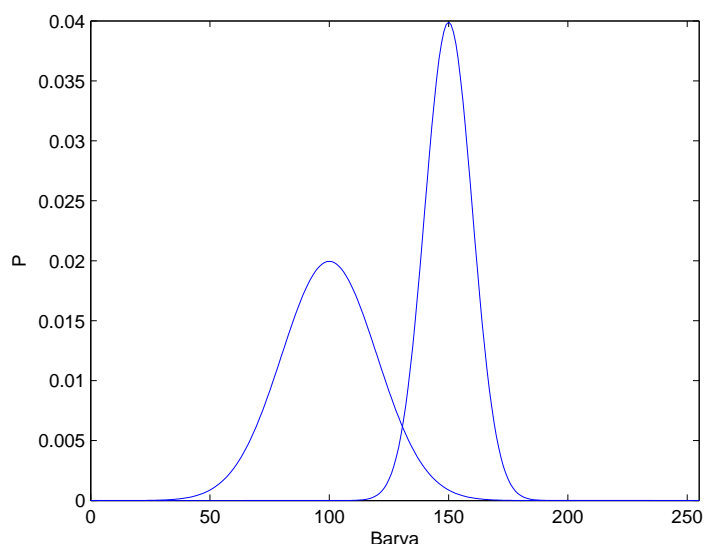
kde

x je zpracovávaný pixel daného obrazu,

μ_n značí střední hodnotu $\mathbf{E}(X)$ n -tého pozadí,

σ_n je směrodatná odchylka

a σ_n^2 je rozptyl pravděpodobnosti $\mathbf{D}(X)$ n -tého povrchu.



Obr. 1.1: Gaussovo rozdělení pravděpodobnosti

Například z obrázku 1.1 je patrné, že se hodnoty pixelu pohybují kolem středních hodnot 100 a 150, což v tomto případě odpovídá dvěma povrchům. Z důvodu vyšší pravděpodobnosti křivky kolem hodnoty 150 lze pak předpokládat, že právě tato definuje pozadí zpracovávaného pixelu, a nižší křivka může odpovídat objektu, který se pohyboval před pozadím.

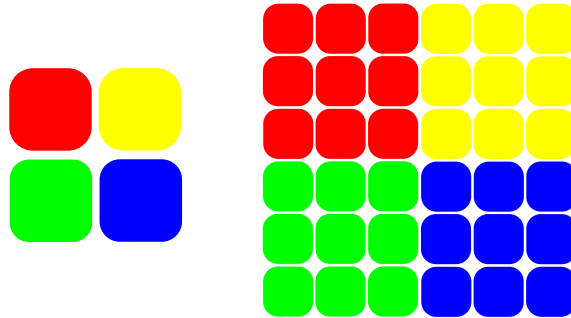
Metoda MOG nachází své uplatnění v místech, kde dochází k periodickým změnám pozadí, například během chvění větví ve větru může jeden pixel zobrazovat

větve, budovu za stromem či oblohu. Dochází zde také k průběžné aktualizaci modelu pozadí. Metoda tedy dokáže průběžně reagovat na dynamické změny pozadí, čímž je již vhodná pro použití v reálném čase. [2], [4], [6].

1.5 Samoorganizační neuronové sítě

Odlišný přístup pro vytvoření modelu pozadí umožňuje použití neuronových sítí. Samoorganizační sítě se vyznačují automatickým učením bez učitele, kdy sítě na základě daných vztahů reagují na vstupní data správnou informací na výstupu.

Sítě jsou koncipovány jako dvourozměrné uspořádání uzlů, neuronů, kdy každému pixelu odpovídá jedna $n \times n$ rozměrná síť. Pro každý uzel dané sítě je vypočtena funkce vážených lineárních kombinací, které jsou přítomny na vstupu sítě. Každý uzel lze tedy popsat pomocí váhového vektoru a kombinace všech váhových vektorů pak definuje model pozadí B . Počáteční model pozadí B^0 je pak stanoven z prvního snímku ve videu, kde hodnoty jednotlivých váhových vektorů v $n \times n$ maticích korespondují s hodnotami příslušného pixelu ve snímku. Příklad pro $n = 3$ je uveden na obrázku 1.2.



Obr. 1.2: Obrázek (vlevo) a příslušná neuronová síť (vpravo)

Učení probíhá takzvanou soutěžní strategií. Po předložení vzoru, tedy pixelů dalšího snímku na vstupy sítě, dojde k soutěži, která spočívá v porovnání euklidovských vzdáleností mezi vstupním vzorem a neurony v síti. Vítězným neuronem se zde stává ten, jenž má euklidovskou vzdálenost nejmenší. Souběžně s tímto procesem je vytvořen snímek vzdáleností D^t , pomocí něhož je postupným porovnáváním dle vztahu 1.6 s hodnotou prahu ϵ získána maska popředí. Model pozadí je vzápětí určen aktualizací váhových vektorů určených z předchozího snímku.

$$D^t > \epsilon \quad (1.6)$$

Dle výzkumů provedených v [7] dosahovala metoda poměrně dobrých výsledků. Úspěšně se adaptovala v případech s chvějícími se větvemi, vodní fontánkou či ve videosekvencích s mořskými vlnami. Výpočetní složitost této metody je $O(n^2 \cdot N \times M)$, kde n^2 je počet váhových vektorů pixelu a $N \times M$ jsou rozměry videa. Metoda je v závislosti na velikosti n vhodná pro použití v reálném čase pouze do určitého rozlišení videa. [7]

1.5.1 MIL-SOBS

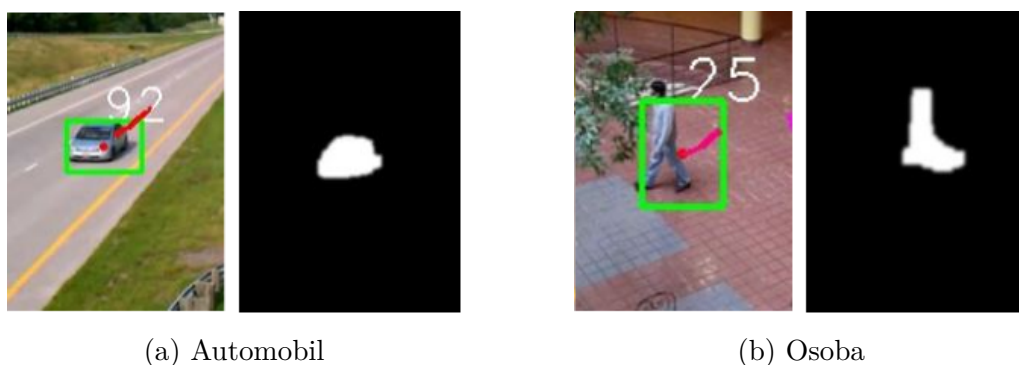
Pro dosažení multimodálnosti byla metoda modifikována o možnost použití více vrstev neuronových sítí. Modifikovaná metoda se nazývá MIL-SOBS, název pochází z anglického Multi Independent-Layered Self-Organizing Background Subtraction. Pro každou vrstvu je, obdobně jako u jednovrstvého případu, určen model pozadí. Vrstvy na sobě nejsou závislé, takže změna v jedné vrstvě neovlivní vrstvy další. Tato nezávislost má za následek řadu nezávislých modelů pozadí, které se liší například v osvětlení scény, poloze kamery apod. Model pozadí je adaptován pouze pro tu vrstvu, ve které byly určeny nejnižší euklidovské vzdálenosti.

Metoda byla ještě doplněna o možnost automatického vyvažování pomocí distribučních funkcí Gaussova rozdělení pravděpodobnosti, čímž dochází k průběžnému sledování dynamiky pixelů. Úspěšnosti metod byly porovnány a shrnuty v literatuře [8]. Metody byly testovány na videosekvencích obsahujících mnoho dynamických prvků, chybovost kamer, stíny apod. Metoda MIL-SOBS se díky vícevrstvé struktuře prokázala být vhodnou pro použití ve venkovním prostředí, nižších úspěšností však dosahovala v momentě, kdy objekt, který byl přítomen ve fázi tréninku, opustil scénu, např. když auto, které stálo na parkovišti, odjelo. [8], [9].

2 DETEKCE A ZNAČENÍ OBJEKTŮ

Úspěšným odstraněním pozadí a získáním masky popředí lze získat dobrou představu o výskytu pohyblivých objektů před kamerou. Hlavní úlohou algoritmů pro detekci objektů je určit velikost daného objektu, jeho tvar a trajektorii pohybu. Je také vhodné správně odhadnout, jakým směrem se bude objekt dále pohybovat, čemuž se věnuje text v kapitole 3. Je-li to v dané aplikaci vyžadováno, je také algoritmus doplněn o možnost klasifikace. Algoritmus pak musí správně vyhodnotit, jedná-li se například o osobní automobil, motocykl, chodce apod. Ne vždy se však, z důvodu slabší účinnosti algoritmů pro eliminaci pozadí, podaří získat obrys celého objektu, což má za následek chyby v klasifikaci.

Tato kapitola se zabývá přístupy, které mají za úkol určit z takzvaného blobu (blob – Binary Large Object), což je obrys neznámého objektu, který se nachází v masce popředí, tvar celkového objektu, a poté jej správně ohraničit. Příklad ohraničení a blobu je pro automobil uveden na obrázku 2.1a a pro osobu na obrázku 2.1b. Existují čtyři typy těchto přístupů. Je však důležité zmínit, že algoritmy z těchto kategorií je možné pro zlepšení výsledné účinnosti kombinovat. [10], [11].



Obr. 2.1: Detekce a blob

2.1 Region

Region lze v obraze definovat jako ohraničené okolí blobu získaného předchozím zpracováním. Objekt je ohraničen mnohoúhelníkem, který má co nejmenší počet vrcholů. Region je poté reprezentován souřadnicemi vrcholů tohoto mnohoúhelníku, což má za následek nižší výpočetní náročnost. Z tohoto důvodu však region přesně nekopíruje hranice daného objektu, což je v řadě aplikací žádoucí. Tvar a počet souřadnic regionu zůstává nezměněn po dobu pohybu objektu ve scéně za předpokladu,

že nedochází k výrazným změnám obrysu objektu. Změna regionu je při změně obrysu, například výskyt objektu ve scéně pod jiným úhlem, zajištěna potřebnými geometrickými úpravami, jako je rotace, či translace. [12], [13].

2.2 Kontura

Algoritmy pracující s konturami mají za cíl přesné ohraničení objektu, čímž poskytují přesnější informace o jeho tvaru než algoritmy založené na regionech (bílá plocha na obrázcích 2.1a a 2.1b). V tomto případě je ohraničen přímo obrys objektu. V každém snímku jsou pak získány kontury nové, neboť s každým dalším snímkem dochází ke změnám obrysů objektů. [10], [12], [14].

2.3 Příznaky

Algoritmy založené na používání příznaků pracují se segmenty, které jsou extrahovány z obrazu. Segmenty jsou sloučeny do vysokoúrovňových příznaků, a příznaky jsou poté přiřazovány mezi snímky. Příznakové algoritmy mohou být rozděleny do těchto tří kategorií:

- globální – plocha, barva, střed objektu apod.,
- lokální – segmenty čar, křivek, souřadnice rohů,
- závislostní – vzdálenosti a geometrické vztahy mezi příznaky.

Algoritmy z výše zmíněných kategorií mohou být pro zkvalitnění výsledků kombinovány. U příznakových algoritmů dochází ve dvourozměrném obraze k úspěšné a rychlé adaptaci, což umožňuje použití v reálném čase. Pro závislostní algoritmy to však neplatí, neboť zde probíhá časově náročné hledání a porovnávání závislostí.

Úspěšnost rozpoznání objektů založených na 2D příznacích je však bohužel velmi nízká, z důvodů nelineárních zkreslení během perspektivní projekce, či kvůli pohybu snímacího zařízení. [12]

2.4 Model

Modelem se zde rozumí objekt, který je vytvořen na základě předchozích znalostí objektů, jež se vyskytují v obraze. Modely jsou nejčastěji tvořeny pomocí manuálního nastavování, nástrojů CAD či technik počítačového vidění.

Pro sledování pohybu osob je použit základní přístup analýza syntézou. Nejprve je z předchozího pohybu predikována základní poloha postavy pro následující snímek. Následně je predikovaný model syntetizován a porovnán se skutečnými daty z tohoto snímku, čímž je určena podobnost mezi těmito daty. Toho lze v závislosti

na použitém algoritmu dosáhnout buďto rekurzivně, nebo pomocí vzorkovacích technik, dokud není nalezena správná poloha postavy a model může být aktualizován. Definici modelu pro první snímek je nutné řešit individuálně. [12], [15].

2.4.1 Model postavy

Na úspěšnosti trasovacích algoritmů se zde největší měrou podílí kvalita vytvořeného modelu postavy. Čím je model komplexnější, tím jsou výsledky přesnější, avšak za cenu vyšší výpočetní náročnosti. Lidská postava může být reprezentována následujícími přístupy. [12]

Stick figure

Lidský pohyb je charakteristický pohybem končetin, těla a hlavy, které jsou spojeny klouby. Stick figure je nejjednodušší metodou reprezentace modelu člověka, neboť jak tělo, tak i končetiny jsou vyjádřeny jednoduchými liniemi. [12]

2D kontury

Model lidské postavy vytvořený 2D konturami přímo odpovídá obrysům lidského těla v obraze. Části lidského těla jsou v tomto případě reprezentovány vzájemně propojenými dvourozměrnými pásy. Výhodou je již přesnější zachycení celé postavy, modely jsou ale již komplikovanější. [12]

3D modely

Hlavní nevýhodou předchozích dvou modelů je, že nereflektují různé úhly natočení přítomné postavy. Tuto nevýhodu se daří odstranit použitím nejrůznějších trojrozměrných prvků, jimiž jsou například koule, super kvadriky, eliptické válce či kužely. Použití trojrozměrných prvků však vyžaduje také více parametrů, čímž dochází ke značnému zvýšení výpočetní náročnosti. [12]

Hierarchický model

Pro zpřesnění výsledků byl také navržen komplexní hierarchický model, který již téměř kopíruje strukturu lidského těla. Základní podstatou hierarchie je kostra, kolem které se nachází eliptické válce značící lidské svalstvo. Svaly jsou potaženy mnohoúhelníkovým povrchem symbolizujícím kůži. Z důvodu vysoké komplikovanosti takových modelů již není možné použití v reálném čase. Kvalitnější modely však nacházejí uplatnění například ve zdravotnictví. [12]

2.4.2 Modely pohybu

Modely pohybu nacházejí v oblasti sledování široké uplatnění. Je tomu tak z toho důvodu, že lidské tělo umožňuje vykonávat předem známé pohyby limitované omezením lidských kloubů. Mezi takové pohyby patří chůze, běh, mávání, hýbání trupem apod. Modely pohybu slouží jako základní znalost pro následnou predikci pohybu ve scéně, pro rozpoznání lidského chování, či pro usnadnění výpočtů při modelování postavy.

Nejdůležitější vlastností modelů pohybu je jejich přirozenost. V závislosti na prostředí a vlivech okolí může docházet k rozdílnému pohybu byť stejné osoby. Velké odlišnosti mohou také nastat při porovnání pohybu dvou různých osob. Každý člověk má rozdílný styl chůze, například mladý člověk jde rychleji než starší osoba s holí. Přirozenost modelu spočívá v adaptaci modelu na pohybové vlastnosti dané osoby v prostředí, ve kterém se právě nachází. Toho lze dosáhnout ve třech hlavních krocích:

1. Analýza pohybu – manuální volba snímků s významnými momenty scény (např. došlápnutí, úkrok . . .), zohlednění vlivů prostředí, výpočet příznakových vektorů pro vybrané snímky.
2. Modelování pohybu – z předchozích vektorů pomocí Gaussova rozdělení, rozdělení na časově proměnné a neproměnné vlastnosti nebo jedná-li se o pohyb periodický, či nikoliv.
3. Odhad parametrů – úprava parametrů předchozího statistického modelu. [12].

3 SLEDOVÁNÍ A PREDIKCE POHYBU

Pohyb osob či vozidel ve snímcích ze statických kamer lze označit za náhodný, stochastický proces. Z toho důvodu je odhad následující polohy objektu ve vícerozměrné scéně značně obtížný. Nejvyužívanějšími metodami pro sledování a odhad následujícího pohybu je Kalmanův filtr a metoda sledování optického toku.

3.1 Kalmanův filtr

Úkolem Kalmanova filtru (KF) je na základě předchozího pohybu objektů odhadnout jejich následující polohu. Vyvinut byl již v letech 1960–1961 maďarským vědcem Rudolfem Kálmánem za účelem filtrace šumu z elektrických signálů. Své uplatnění našel v širokém spektru oborů, jako například letectví, kdy se významnou měrou podílel na vývoji autopilotů letadel. Jedná se o filtr dynamický. Charakteristickou vlastností je pro něj znalost modelu systému, což mu umožňuje predikovat následující stav. Predikce je úspěšná dokonce i v datech zatížených nepřesnostmi či šumem. Filtr pracuje cyklicky ve dvou fázích. Jsou jimi fáze predikční a fáze měření.

Predikční fáze má na starosti předpověď následujícího stavu a aktualizaci chyby měření. Následující fáze zajišťuje porovnání predikce se skutečnými naměřenými daty a predikci opravuje. Predikce následujícího stavu probíhá dle vztahu (3.1) a odhad chyby dle vztahu (3.2).

$$\hat{x}_k^- = \mathbf{A}\hat{x}_{k-1} + w_{k-1}, \quad (3.1)$$

$$P_k^- = \mathbf{A}P_{k-1}\mathbf{A}^T + \mathbf{Q}, \quad (3.2)$$

kde \hat{x}_k^- je predikovaný stav filtru pro současný stav k ,

P_k^- je odhad chyby pro daný stav,

\mathbf{A} je $n \times n$ rozměrná přechodová matice značící vztah mezi předchozím ($k - 1$) a současným (k) stavem systému,

w_{k-1} je hodnota procesního šumu,

P_{k-1} je předchozí chyba výpočtu

a \mathbf{Q} značí kovarianční matici procesního šumu.

V následující fázi je objekt vyhledáván a na základě rozdílu mezi predikcí a skutečností je odhad aktualizován dle vztahu (3.3) přičtením Kalmanova zisku K_k ovlivněného měřením z_k (vztahy 3.4 a 3.5).

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-), \quad (3.3)$$

$$K_k = \frac{P_k^- \mathbf{H}^T}{(\mathbf{H}P_k^- \mathbf{H}^T + \mathbf{R})}, \quad (3.4)$$

$$z_k = \mathbf{H}x_k + v_k, \quad (3.5)$$

kde \mathbf{H} je $n \times m$ rozměrná matice měření vztažená k naměřené hodnotě \hat{x}_k a \mathbf{R} je $m \times m$ rozměrná matice šumu v_k vzniklého při měření. Na základě hodnot Kalmanova zisku a matice měření poté dojde k aktualizaci chyby měření P_k .

$$P_k = (\mathbf{I} - K_k \mathbf{H}_k) P_k^-, \quad (3.6)$$

kde \mathbf{I} je v tomto případě jednotková matice. Po měření a úpravě všech predikovaných hodnot dojde k předání parametrů do predikční fáze a proces probíhá znovu pro další snímek. [16], [19], [17].

3.2 Optický tok

Metoda optický tok přistupuje k problému trasování objektů zcela odlišným způsobem než tomu bylo v předchozím případě. Každému pixelu je přiřazena rychlost, neboli vzdálenost, kterou pixel absolvoval mezi dvěma po sobě jdoucími snímky. Rychlost se skládá ze dvou složek, složka rychlosti u , pro směr dle osy x , a složka v , pro pohyb ve směru osy y . Pro takový pixel se předpokládá, že hodnota jeho jasu je v novém snímku shodná s předchozím, tudíž platí následující vztah:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t), \quad (3.7)$$

kde $I(x, y, t)$ je funkce jasu v daném bodě a čase a $I(x + \Delta x, y + \Delta y, t + \Delta t)$ je tato funkce v následujícím časovém kroku, s novou polohou daného pixelu. Úpravou předchozího vztahu je pak získána základní pohybová diferenciální rovnice optického toku, ve které jsou neznámými rychlostní složky u a v :

$$\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} = 0 \quad (3.8)$$

kde $\frac{\partial I}{\partial x}$ značí parciální derivaci jasové funkce podle x , respektive $\frac{\partial I}{\partial y}$ podle y .

Základním omezením při určování optického toku je takzvaný aperturový problém. Ten spočívá v tom, že není možné určit pohybovou složku, která je kolmá na

směr gradientu příslušné hrany. Nelze tedy správně určit pohyb objektu, který se pohybuje rovnoběžně se směrem jeho hrany. Například v případě, že se přímka viditelná skrz otvor pohybuje směrem nahoru. V tomto případě může pozorovatel nabýt dojmu, že zde k žádnému pohybu nedochází. Pro určení optického toku je tedy nutné získat složky u a v . Pro získání těchto složek jsou v současnosti používány přístupy pracující s hustým a řídkým optickým tokem. V případě hustého optického toku je určen optický tok pro každý pixel daného snímku. Typickým představitelem je Horn-Schunckova metoda. Z důvodu vyšší výpočetní náročnosti předchozího přístupu byl zaveden přístup zvaný řídký optický tok, který předem specifikuje požadované pixely, které mají být sledovány v nastaveném vyhledávacím okně. Těmito body jsou například hrany či rohy objektů, díky čemuž dochází k úspěšnějšímu a spolehlivějšímu sledování objektů. Nejpoužívanější metodou využívající řídkého optického toku je metoda Lucas-Kanade. Zmiňované metody jsou blíže popsány v následujících podkapitolách.[18], [19], [20].

3.2.1 Horn-Schunck

Metoda Horn-Schnuck umožňuje řešení rovnice (3.8) díky předpokladu, že sousední body ve scéně odpovídají stejnému objektu a konají obdobný pohyb. Míru, s jakou se liší tento předpoklad od skutečnosti, lze popsat vztahem (3.9):

$$S = \int_0^N \int_0^M \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 dx dy, \quad (3.9)$$

kde N je výška a M je šířka zpracovávaného obrazu. Jednotlivé členy vztahu vyjadřují změnu příslušné složky rychlosti v daném směru. Pro dosažení platnosti výše zmiňovaného předpokladu je hledána nejnižší hodnota parametru S . Platnost vztahu (3.8) je ověřována v následujícím vztahu, který by měl v ideálním případě nabývat nulových hodnot.

$$C = \int_0^N \int_0^M \left(\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \right)^2 dx dy, \quad (3.10)$$

Vzhledem k přítomnému šumu a kvantizační chybě však hodnota C nebude v reálné situaci s nejvyšší pravděpodobností nikdy nulová. Z tohoto důvodu je zaveden parametr λ , který zohledňuje úroveň přítomného šumu. Za účelem získání hodnoty optického toku je pak hledána nejnižší hodnota následujícího vztahu:

$$\mathcal{E} = S + \lambda C. \quad (3.11)$$

Vhodnými úpravami předchozích vztahů jsou pak získány následující rovnice, které již umožňují vypočítat hodnoty pohybových složek u a v a určit tak optický tok.[19], [20].

$$\begin{aligned} I_x^2 u + I_x I_y v &= \lambda \nabla^2 u - I_x I_t, \\ I_x I_y u + I_y^2 v &= \lambda \nabla^2 v - I_y I_t. \end{aligned} \quad (3.12)$$

3.2.2 Lucas-Kanade

Metoda Lucas-Kanade (LK) vychází z předpokladu konstantního jasu mezi dvojicí snímků u daného objektu. Metoda je doplněna o předpoklad, že v malém okně kolem jednoho pixelu jsou rychlostní složky konstantní. Pro $m \times m$ rozměrné okno lze tedy stanovit $n = m^2$ rovnic s dvojicí neznámých, které vyplývají ze vztahu (eq:tok):

$$\begin{aligned} \frac{\partial I}{\partial x_1} u + \frac{\partial I}{\partial y_1} v &= -\frac{\partial I}{\partial t_1}, \\ \frac{\partial I}{\partial x_2} u + \frac{\partial I}{\partial y_2} v &= -\frac{\partial I}{\partial t_2}, \\ &\vdots \\ \frac{\partial I}{\partial x_n} u + \frac{\partial I}{\partial y_n} v &= -\frac{\partial I}{\partial t_n}. \end{aligned} \quad (3.13)$$

Pro zlepšení výsledků použití algoritmu LK byla představena pyramidická verze tohoto algoritmu. Obraz je několikrát podvzorkován, každá vyšší vrstva má poloviční šířku a poloviční výšku. Určování optického toku pak probíhá od nejvyšší vrstvy, kde je rozpoznán ten nejvýraznější pohyb ve scéně. Odhad pohybu z předchozí úrovně je považován za počáteční bod pro odhad pohybu pro následující vrstvu. [18], [19], [20].

4 NÁVRH A IMPLEMENTACE

Aplikace z části vychází z práce Kim Dinh Sona z Vietnamu, která již v jistých případech vykazuje poměrně dobrou úspěšnost.[21] Tato práce byla doplněna o algoritmus pro eliminaci stříhu, o algoritmus sledující pohyb ve vybrané, libovolně volitelné části scény a také bylo vytvořeno grafické rozhraní umožňující jednoduché nastavení parametrů i pro méně zkušeného uživatele. Tato aplikace byla vytvořena a implementována v jazyce Java pomocí knihovny OpenCV. Jako vývojové prostředí byl použit program Eclipse ve verzi Mars.1 (4.5.1).

4.1 Knihovna OpenCV

Knihovna OpenCV je balík softwarových nástrojů, které umožňují snadnou aplikaci algoritmů počítačového vidění. Celým názvem Open source Computer Vision Library byla představena již v roce 1999 firmou Intel. Jak již název napovídá, jedná se o knihovnu s otevřeným kódem pod licencí BSD. Tato licence umožňuje volné použití a modifikaci dostupných kódů jak v soukromé či akademické, tak i v komerční sféře.

Zdrojový kód byl naprogramován v jazycích C a C++, nicméně algoritmy lze používat i v řadě dalších programovacích jazyků, jako jsou například Java či Python. Samozřejmostí je také podpora řady nejrozšířenějších operačních systémů, jako jsou Windows, Linux, Android či Mac OS. Knihovna je v neustálém vývoji, v současnosti je v ní více než 2500 optimalizovaných algoritmů rozdělených do několika modulů v závislosti na možnosti použití.

Nejdůležitějším modulem je modul `core`, který obsahuje definice základních struktur. Snímek je ukládán ve formátu matice, lze pracovat s více bitovými hloubkami v celočíselném či desetinném zpracování. Pro více funkcí je však nutné implementovat funkce z více modulů. Modul `ImgProc` (Image Processing) umožňuje používat geometrické transformace, filtrování obrazu, histogramy a mnoho dalšího. Modul `videoio` zpřístupňuje přehrávání videa a v modulu `video` jsou dostupné algoritmy pro zpracování videosekvencí, například pro odstranění pozadí. V knihovně se nachází velké množství dalších modulů, pro trasování objektů jsou však důležité výše uvedené.

V této práci je použita knihovna ve verzi 3.1, která je dostupná od 21. 12. 2015, souběžně s touto verzí však stále probíhá vývoj knihovny řady 2.4.x, nejaktuálnější verze 2.4.13.2 byla zveřejněna 16. 12. 2016.

Aplikace se skládá z grafické a obslužné části. Grafické rozhraní sestává z trojice oken, jedno zobrazuje vstupní video, druhé okno slouží pro zobrazení masky popředí

a třetí okno umožňuje nastavení parametrů. Obslužná část se stará o správný chod aplikace. [19], [22].

4.2 Výsledná aplikace

Spuštění probíhá buďto přímo z vývojového prostředí, nebo pomocí příkazového řádku zadáním následujícího příkazu pro příslušnou sekvenci:

Výpis 4.1: Spuštění programu

```
java -jar "cislo_sekvence.jar"
```

Výsledný program se skládá z jedenácti tříd rozdělených do dvou balíčků. Struktura je uvedena ve výpisu 4.2.

Výpis 4.2: Balíčky a třídy

```
cz.vutbr.feec.objtracking
- CONFIG.java
- Functions.java
- GUI.java
- MainApp.java
- Zone.java
cz.vutbr.feec.objtracking.predikce
- AssignmentOptimal3.java
- HungarianAlg3.java
- JTracker.java
- Kalman.java
- Track.java
- Tracker.java
```

Balíček `cz.vutbr.feec.objtracking` zapouzdřuje třídy na obsluhu grafického rozhraní, eliminaci pozadí a na správu zón a hranic. Třída `CONFIG` obsahuje cestu k zpracovávanému souboru a parametry ovlivňující detekční a predikční schopnosti algoritmu. Tyto parametry je možné měnit pomocí grafického rozhraní a více je jim věnována kapitola 4.3.

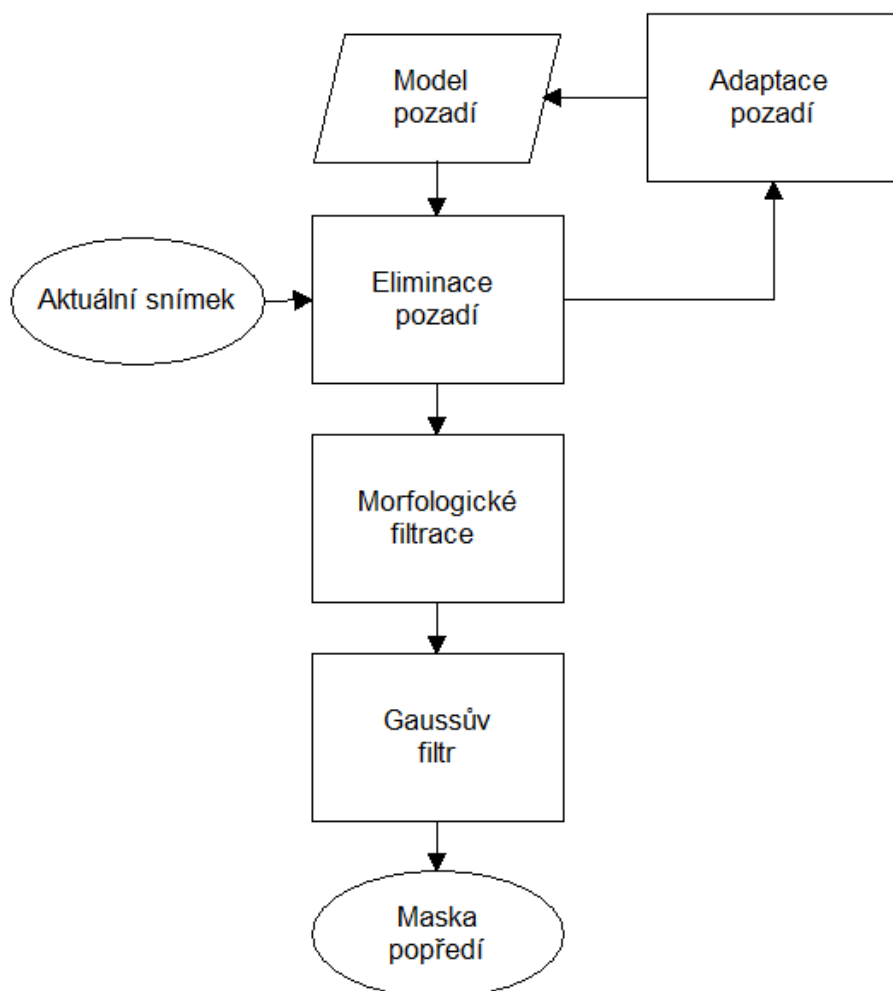
Ve třídě `Functions` se nachází funkce pro odstranění pozadí a získání masky popředí, tomuto postupu je věnována následující podkapitola 4.2.1. Třída `GUI` obsluhuje grafické rozhraní pro editaci parametrů, které je důkladně rozebráno v kapitole 4.3. Třída `MainApp` je spouštěcí třídou aplikace. Tato třída obsluhuje hlavní okno grafického rozhraní, je zde implementován algoritmus na eliminaci stříhu a volá

potřebné metody z ostatních tříd. Třída `Zone` byla vytvořena za účelem práce se zónami. Zóny umožňují monitorovat a zaznamenávat pohyb ve specifikované oblasti. Více je tomuto tématu věnována samostatná podkapitola 4.3.4.

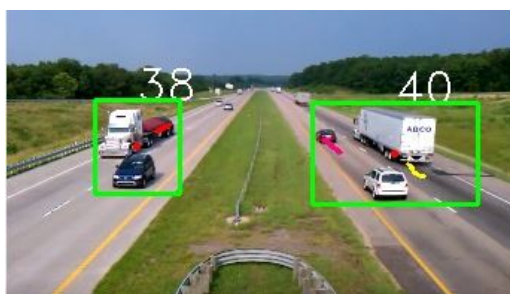
V balíčku `cz.vutbr.feec.objtracking.predikce` se nachází šest tříd pro samotné sledování a predikci pohybu. Funkčnost těchto tříd je rozebrána v kapitole 4.2.2.

4.2.1 Analýza pohybu

Na obrázku 4.1 je vyobrazen postup pro získání masky popředí z aktuálního snímku. Prvním krokem je eliminace pozadí. Pro tyto účely je v modulu `video` v jazyce Java k dispozici algoritmus `BackgroundSubtractorMOG2`, který je založený na metodě Mixture of Gaussians (viz kapitola 1.4). Výstup použité metody je demonstrován na obrázku 4.2.



Obr. 4.1: Blokový diagram eliminace pozadí



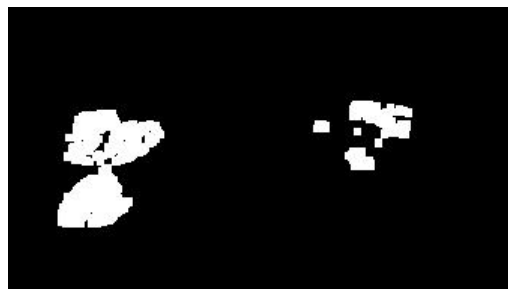
(a) Originální snímek



(b) Aplikovaná metoda MOG



(c) Aplikovaná operace eroze



(d) Aplikovaná operace dilatace



(e) Aplikovaná operace otevření



(f) Aplikovaná operace uzavření

Obr. 4.2: Originální snímek a výsledky morfologických operací

Z obrázku 4.2b je patrné, že výstup metody MOG obsahuje šum, objekty nejsou celistvé a dochází k jejich překrývání. Aplikací řady níže uvedených morfologických operací lze takto vzniklé chyby eliminovat a zvýšit tak úspěšnost implementovaného algoritmu.

Eroze

Morfologickou operací eroze lze docílit odstranění přítomného nežádoucího šumu, za cenu zmenšení žádoucích objektů. Eroze může být definována jako složení dvou množin bodů, s použitím vektorového rozdílu. Jedná se o množiny z masky popředí a strukturní element o rozměrech 5×5 pixelů. Příklad strukturního elementu **SE** je uveden v rovnici (4.1). Strukturní elementy dalších operací vypadají obdobně, liší se nanejvýš rozměrově. Výsledek operace eroze je zobrazen na obrázku 4.2c. [23]

$$\mathbf{SE} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (4.1)$$

Dilatace

Úspěšným odstraněním přítomného šumu došlo ke zmenšení přítomných objektů, v řadě případů se objekty dokonce rozdělily do několika segmentů. Operace dilatace umožňuje rozšířit a spojit dílčí segmenty přítomného objektu. Dilataci lze definovat jako vektorový součet bodů dvou množin. V tomto případě se jedná o strukturní element o rozměrech 5×5 pixelů a masku popředí získanou z předchozího zpracování. Z obrázku 4.2d je patrné, že detekované objekty se rozšířily a souběžně s tím došlo také k jejich propojení do spojitých celků. [23]

Otevření a uzavření

Operace otevření a uzavření jsou kombinací výše zmíněných. Otevření provádí operaci eroze a na erodovaný snímek je poté aplikována operace dilatace. Otevření má za následek odstranění drobnějších detailů. Operace uzavření používá přesně opačný postup a jejím důsledkem je jednodušší obraz. Výsledky těchto operací jsou zobrazeny na obrázcích 4.2f a 4.2e.

Rozmazáním snímků pomocí Gaussova filtru je pak získána výsledná maska popředí (obrázek 4.3). Souběžně s tím se algoritmus Mixture of Gaussians stará o aktualizaci modelu pozadí. Tyto algoritmy jsou implementovány ve funkci `processFrame` třídy `Functions`. [23]



Obr. 4.3: Výsledná maska popředí

Detekce kontur

Z masky popředí jsou pomocí funkce `detectionContours` získány kontury požadovaných objektů a pro každý objekt je pomocí funkce `boundingRect` z modulu `ImgProc` získán obdélník, který ohraničuje celou plochu kontur. Následně jsou určeny středy těchto obdélníků. Objekty jsou označeny zeleným obdélníkem, střed objektu je vyznačen červeně.

4.2.2 Trasování objektů

Pole detekovaných objektů je spolu s jejich získanými středy předáno metodě `update` třídy `Tracker`. Deklarace parametrů třídy `Tracker` se nachází v abstraktní třídě `JTracker`. S každým novým bodem je vytvořena nová trasa. Trasa je definována daným bodem a unikátním označením ID, kterým jsou poté označovány samotné objekty. Parametry trasy obsahuje třída `Track`. Pro úspěšné trasování detekovaného objektu je nutné jej správně přiřadit k vytvořené predikci trasy.

Za účelem trasování objektů se v knihovně OpenCV nachází Kalmanův filtr, pyramidická verze algoritmu LK a metoda optického toku založená na husté reprezentaci. Tyto metody byly diskutovány v kapitole 3. Pro každou novou trasu je vytvořen Kalmanův filtr s následující přechodovou maticí **A** (4.2), kovarianční maticí měřicího šumu **R** (4.3) a kovarianční maticí procesního šumu **Q** (4.4). Pro inicializaci a aktualizaci parametrů KF slouží třída `Kalman`.

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.2)$$

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (4.3)$$

$$\mathbf{Q} = \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 \end{bmatrix}, \quad (4.4)$$

kde Δt je volitelný interval vzorkování, výchozí hodnota je nastavena na 0,1 s.

Spárování trasy s příslušným objektem zajišťují funkce tříd `AssignmentOptimal` a `HungarianAlg3`. V těchto třídách je implementována takzvaná maďarská metoda, která byla vytvořena pro účely řešení přiřazovacích problémů minimalizačního typu.

Řešení přiřazovacího problému spočívá v přidělení n pracovníků k vykonání n požadavků s využitím co nejmenšího počtu zdrojů.

S každým novým snímkem probíhá predikce pohybu a vykreslování trajektorie do hlavního okna programu. Ztratí-li se objekt ze scény z důvodu přítomné překážky, dokáže algoritmus po definovaný počet snímků na základě předchozího pohybu a rychlosti odhadnout kdy dojde k opětovnému výskytu objektu ve scéně. Tím je zamezeno dvojí detekci jednoho objektu.

4.2.3 Eliminace stříhu

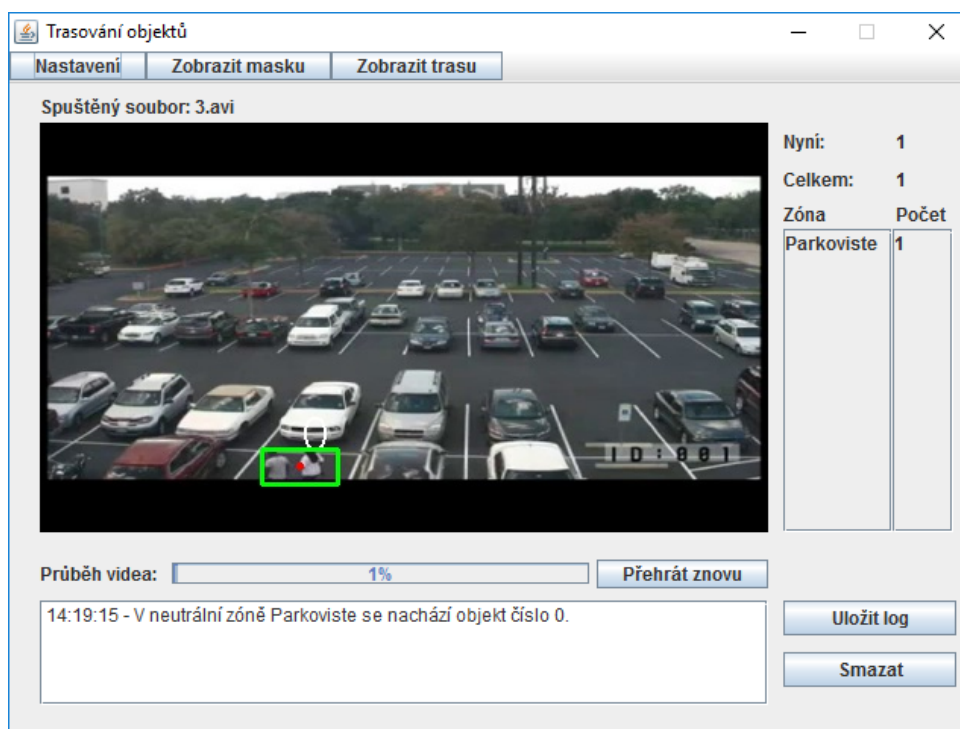
Algoritmus pro eliminaci stříhu vychází z předpokladu, že k výskytu objektů ve scéně dochází pozvolna, nikoli nárazově. Algoritmus zjišťuje počet pixelů v popředí předchozího a aktuálního snímku a určuje jejich rozdíl, který dále porovnává se stanovenou hodnotou prahu. Hodnota prahu je libovolně nastavitelná a odvíjí se od rychlosti a četnosti pohybu a velikosti objektů. V případě, že je rozdíl vyšší než definovaný práh, algoritmus vyhodnotí, že došlo ke stříhu či jiné náhlé změně scény. Následně dojde ke krátkému přerušení trasování a obnovení modelu pozadí, čímž je zamezeno vysokému výskytu falešně pozitivních detekcí.

4.3 Grafické rozhraní

Hlavní okno aplikace (obrázek 4.4) obsahuje především zpracovávanou videosekvenci. Nad videem se nachází informace o jeho názvu. Výběr zpracovávaného videa probíhá zadáním cesty ve třídě `CONFIG`. Tlačítko **Nastavení** spustí okno umožňující úpravu parametrů ovlivňujících výslednou úspěšnost. Toto okno bude podrobněji popsáno v následujících podkapitolách. Tlačítko **Zobrazit masku** vyvolá okno, na kterém se nachází maska popředí pro aktuální snímek sekvence (obrázek 4.5) a tlačítko **Zobrazit trasu** vypíná a zapíná trasu sledovaných objektů.

Na pravé straně od videosekvence je uveden aktuální a celkový počet objektů v dané sekvenci po dobu spuštění aplikace. Dále je zde uvedeno, kolik se za stejnou dobu vyskytlo v dané zóně objektů. Zónám je věnována podkapitola 4.3.4.

V oblasti pod videem je zobrazen ukazatel průběhu videa spolu s možností spuštění jeho opětovného přehrávání. Pod tímto ukazatelem dochází k výpisu detailních informací o pohybu objektů v zónách spolu s časem, ve kterém došlo k výskytu objektu v dané zóně. Tlačítkem **Uložit log** dojde k uložení zobrazených informací do CSV souboru a tlačítko **Smazat** vymaže uvedené informace. Struktura CSV souboru je uvedena v tabulce 4.1, je zde uveden čas, název zóny, typ zóny, unikátní číselné označení objektu a pod těmito informacemi se nachází výčet zón spolu s celkovým počtem objektů v daných zónách zaznamenaných.



Obr. 4.4: Hlavní okno aplikace

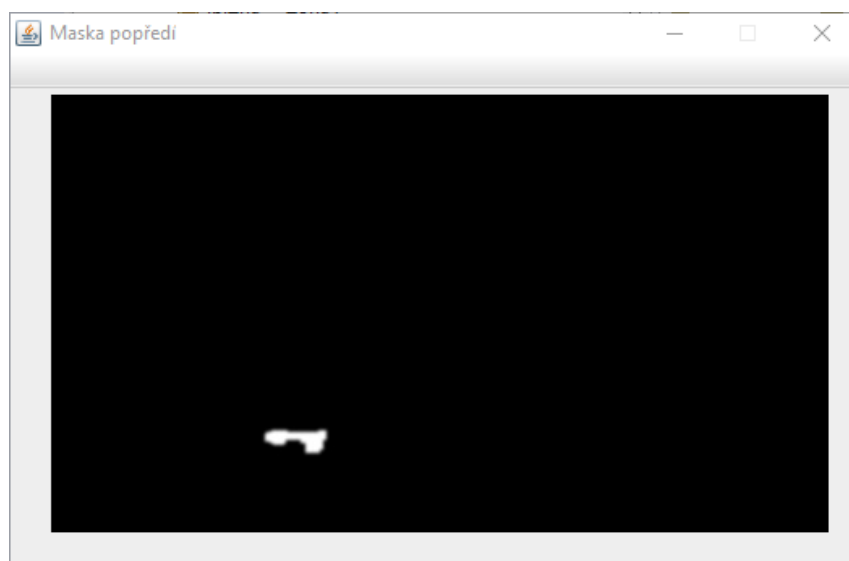
4.3.1 Nastavení

Tlačítko **Nastavení** na hlavním okně otevírá okno umožňující editaci parametrů výrazně ovlivňujících úspěšnost funkčnosti samotného algoritmu (obrázek 4.6). Dominantní část okna patří snímku z aktuální videosekvence. Pod snímek se nacházejí karty **Základní**, **Pokročilé**, **Zóny** a **Hranice**. Pod těmito kartami jsou uvedeny informace o poslední provedené operaci.

4.3.2 Základní

Velikost objektů

V sekci **Velikost objektů** je možné nastavit nejmenší a největší možnou velikost objektu, který bude trasován. Všechny objekty menší, respektive větší než nastavená hodnota budou ignorovány. Nastavení probíhá následovně. Nejprve uživatel zvolí, bude-li nastavovat nejmenší či největší objekt. Současná velikost je zobrazena v levém horním rohu panelu se snímek z videosekvence jako modrý čtverec. Následně uživatel myší vybere požadovanou velikost označenou červeným obdélníkem a tlačítkem **Potvrdit** svoji volbu potvrdí. Nastavení se projeví okamžitě. Vhodnou volbou parametrů lze určit požadovaný typ objektů dané scény. Bude-li ve scéně se společným výskytem chodců a automobilů třeba monitorovat pouze pohyby



Obr. 4.5: Okno s binární maskou popředí

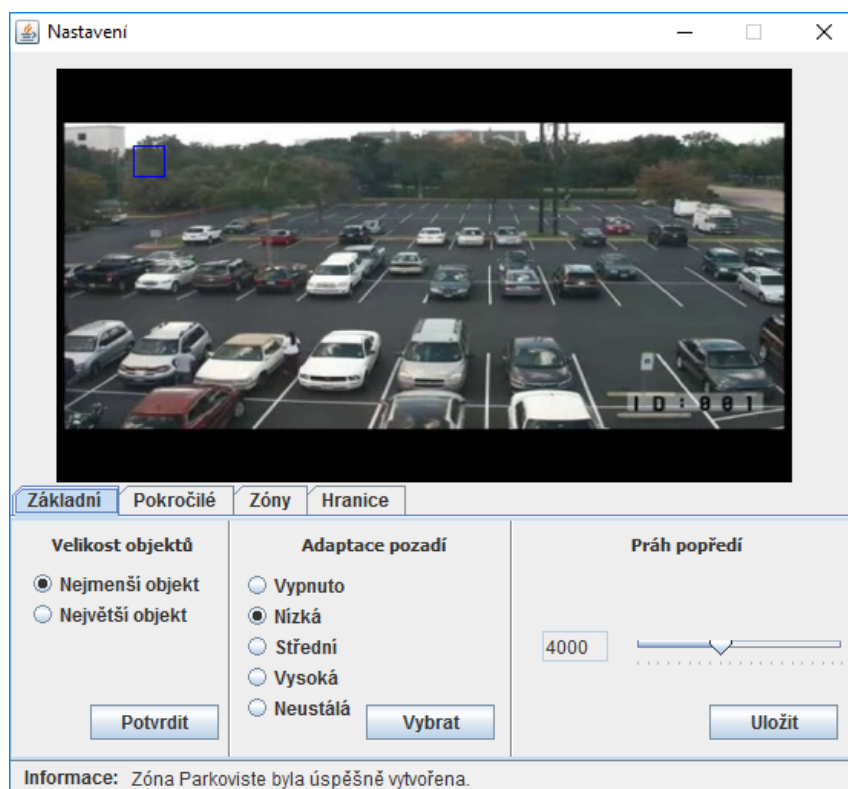
Tab. 4.1: Příklad logu

ČAS	NÁZEV	TYP	ID
14:18:20	Parkoviste	Neutralni	0
14:18:46	Parkoviste	Neutralni	1
14:18:48	Parkoviste	Neutralni	2
14:19:42	Parkoviste	Neutralni	3
14:20:06	Parkoviste	Neutralni	4
NÁZEV	POČET	TYP	
Parkoviste	5	Neutralni	

automobilů, uživatel zvolí pouze rozměry příslušné pro automobily a osoby poté budou ignorovány. Vybrané velikosti jsou reprezentovány jednou číselnou hodnotou. Tato hodnota značí počet pixelů, na nichž se rozprostírá blob daného objektu. Pro maximální možnou velikost je hodnota určena jako obsah zvoleného obdélníku. Pro spodní mez je však nastavena hodnota o 20 % menší, z toho důvodu, že počet pixelů menších objektů bývá zpravidla nižší než je zvolená hranice.

Adaptace pozadí

Sekce **Adaptace pozadí** umožňuje uživateli zvolit rychlost, s jakou se bude model pozadí aktualizovat v závislosti na rychlosti pohybu objektů a dynamických změn ve scéně. Uživatel může volit jednu z pěti následujících možností, kterou poté potvrdí



Obr. 4.6: Okno nastavení – Základní

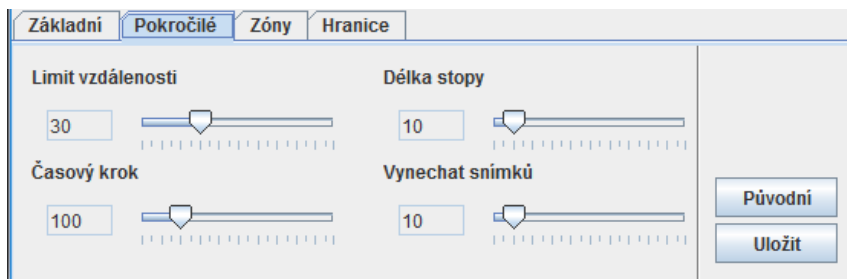
tlačítkem **Vybrat**.

- Volba možnosti **Vypnuto** adaptaci vypne. Tuto možnost je však vhodné zvolit pouze v místech, kde je pozadí statické a nedochází k žádným dynamickým změnám. Například interiér (chodba, ulička v supermarketu apod.).
- Volba nízké adaptace je vhodná v případech, kdy je pohyb pomalý, či pohybují-li se objekty ve směru od nebo ke kameře.
- Střední adaptace je použitelná ve většině případů plynulého pohybu, například záběry z dopravních kamer.
- Vysokou adaptaci je pak možné zvolit v místech s nepřetržitým pohybem, jako mohou být rušné ulice či náměstí ve městech.
- Při zvolené možnosti neustálé adaptace pak nedochází ke sledování žádného pohybu.

4.3.3 Pokročilé

Změna parametrů z obrázku 4.7 přímo ovlivňuje predikční schopnosti Kalmanova filtru. Pro projevení změny je však nutné aplikaci restartovat. Okno je obsluhováno čtveřicí posuvníků s přímým výpisem současné hodnoty do textového pole vlevo od posuvníku. Stiskem tlačítka **Uložit** dojde k uložení nastavených parametrů do

konfiguračního souboru `nazevSekvence.cfg`. Tlačítko **Původní** načte výchozí konfiguraci.



Obr. 4.7: Okno nastavení – Pokročilé

Limit vzdálenosti

Parametr **Limit vzdálenosti** stanovuje maximální možnou vzdálenost mezi predikovanou a skutečnou polohou objektu. Hodnota je uváděna v pixelech. Vzájemná vzdálenost je dána euklidovskou metrikou dle vztahu (4.5).

$$D_e(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}, \quad (4.5)$$

kde $D_e(p, q)$ je funkce euklidovské metriky udávající sumu mocněných rozdílů bodů p a q ,

p_x, p_y jsou souřadnice predikovaného bodu,

a q_x, q_y jsou skutečné souřadnice objektu.

Výchozí hodnota je nastavena na $D_e = 30$ pixelů. V případě příliš nízké hodnoty dochází k nežádoucím vícečetným detekcím jednoho objektu. Pokud je nastavená hodnota příliš vysoká, může docházet k chybám predikce, čímž může dojít k záměně ID mezi více objekty.

Délka stopy

Parametrem **Délka stopy** je určena maximální délka trasy objektu. Při nastavené hodnotě 0 nebude trasa vykreslována, při nastavení maximální hodnoty je trasa vykreslována po celé scéně. Při prvním spuštění daného videa je nastavena výchozí hodnota 10.

Časový krok

Proměnná **Časový krok** přímo ovlivňuje kovarianční matici procesního šumu \mathbf{Q} Kalmanova filtru (4.4) parametrem Δt . Je-li nastavená hodnota příliš nízká, dochází k pomalé konvergenci Kalmanova filtru. To má za následek nepřesnou predikci

a může vyústit i v novou detekci již přítomného objektu. Se zvyšováním hodnoty se predikce přibližují skutečnosti. S vysokou hodnotou však roste citlivost filtru a v případě opětovného výskytu objektu nemusí dojít ke správnému navázání na již existující trasu. Výchozí hodnota je nastavena na $\Delta t = 0,1$ s. S touto hodnotou se výše uvedené nedostatky neprojevují.

Vynechat snímků

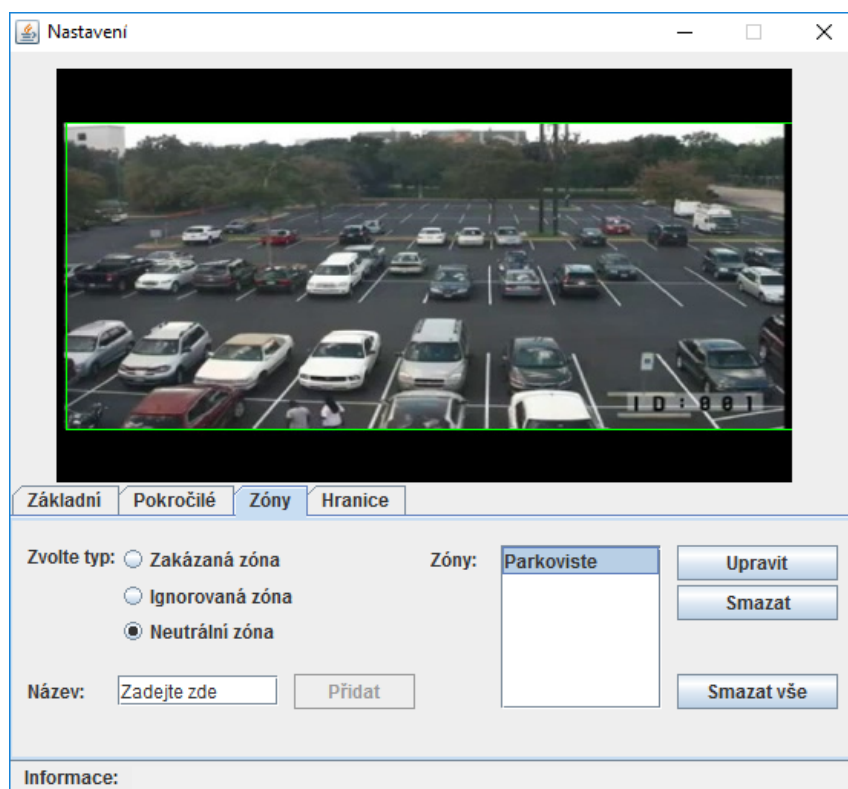
Tento parametr upravuje schopnost Kalmanova filtru navázat na trasu objektu poté, co se objekt ztratil za překážkou a následně se opět objevil ve scéně. Parametr stanovuje kolik snímků lze maximálně vynechat, než bude daná trasa odebrána ze scény. Parametr je vhodné volit v závislosti na rychlosti pohybu objektů a počtu či velikosti různých překážek. V případě, že se v dané sekvenci pohybují objekty rychle, není třeba nastavovat parametr příliš vysoký, neboť lze předpokládat, že se objekt v obraze opět objeví za menší počet snímků než by tomu bylo v případě pomalého pohybu. Příliš nízká hodnota má však za následek duplicitní detekci. Původní trasa objektu je odebrána dříve, než se sledovaný objekt opětovně objeví ve scéně. Jakmile se však objekt dostane opět do popředí, je tento objekt nově označen a je mu predikována nová trasa. Nastavením naopak příliš vysoké hodnoty může dojít ke zbytečné predikci a vykreslování trasy i u objektů, u kterých již opětovný výskyt ve scéně není předpokládán. Například vejde-li osoba do obchodu, či odbočí-li automobil za roh.

4.3.4 Zóny

Karta zóny (obrázek 4.8) umožňuje nastavení uživatelsky definovatelných oblastí obdélníkového tvaru. V případě, že je v některé z těchto definovaných oblastí detekován pohyb, je o tom uživatel informován odpovídajícím způsobem. Zóny jsou zde trojího typu:

- Zakázaná zóna – zóna, kde je výskyt pohybu nežádoucí, např. prostor u expozic v muzeu, v takovém případě dojde k vyvolání poplachu a uložení záznamu scény do složky `alarms`.
- Ignorovaná zóna – tato zóna není pro danou instalaci důležitá, pohyb v ní tedy nebude sledován. Například nebe u záznamů z dopravních kamer.
- Neutrální zóna – jakákoli oblast, ve které je sledovaný pohyb povolen.

Nastavení zón probíhá obdobným způsobem jako tomu bylo u nastavování hraničních velikostí. Uživatel vybere oblast, kterou má daná zóna pokrývat, zvolí její typ a zadá název zóny bez diakritiky. Zóna je vytvořena po stisku tlačítka **Přidat**. Název zóny je unikátní identifikátor, není tedy možné mít dvě zóny se stejným názvem. Zóny se mohou různě překrývat, což však bude mít za následek vícečetné detekce.



Obr. 4.8: Okno nastavení – Zóny

Existující zóny pro danou videosekvenci jsou vypsané v pravé části okna. Výběrem požadované zóny dojde k vykreslení jejích hranic do obrazu zeleným obdélníkem. Tlačítko **Smazat** vymaže vybranou zónu a tlačítko **Smazat vše** smaže všechny. Po stisku tlačítka **Upravit** je možné znovu vybrat hranice zóny a následně je uložit tlačítkem **Potvrdit**.

Pro účely správy zón byla vytvořena třída **Zone**. Třída obsahuje všechny metody potřebné pro práci se zónami. Přidávání zón obsluhuje metoda **add()**. Jejími vstupními parametry jsou souřadnice levého horního a pravého dolního rohu zóny, její název a typ. Tyto parametry jsou zde ukládány do textového souboru nazvaného **nazevSouboru.zone**. Při každé editaci zón a při spuštění programu je volána metoda **loadZones()**. Metoda cyklicky, řádek po řádku, prochází konfigurační soubor a dostupné parametry načítá do trojice proměnných typu **ArrayList<>** pro název, hranice a typ zóny.

S každým novým snímkem pak algoritmus zjišťuje pomocí metody **apply()** výskyt objektu v příslušné zóně. Vstupními parametry pro tuto metodu jsou ID objektu a bod, pro který bude testována příslušnost k některé ze zón. Algoritmus v případě pozitivní detekce prochází cyklicky seznam zón a sleduje, v jaké zóně se testovaný bod nachází a o jaký typ se jedná.

4.3.5 Hranice

V případě, že dochází k pohybu dvou blízkých objektů, které se pohybují téměř stejnou rychlostí, jsou takové objekty detekovány jako jeden objekt. Je-li pohyb těchto objektů přímočarý a z dané scény lze vypořizovat, jakým směrem se budou objekty pohybovat, lze definovat hranice, které sloučenou detekci rozdělí. Typickými představiteli takovéto scény jsou záznamy z dopravních kamer nad víceproutými silnicemi. V takovém případě je hranice přímka, která rozdělí silnici na jednotlivé pruhy.

Okno pro přidávání hranic je zobrazeno na obrázku 4.9. Hranici lze přidat obdobným způsobem jakým probíhalo přidávání zón a vybírání mezních velikostí objektů. Uživatel přímkou vybere oblasti, které potřebuje rozdělít, a stiskem tlačítka **Přidat** dojde k jejich přidání. Souřadnice existujících hranic jsou uvedeny v seznamu a uloženy v souboru `nazevSouboru.brd` ve složce `borders`. Vytvořenou hranici lze po jejím výběru zobrazit v okně a stiskem tlačítka **Smazat** dojde k jejímu smazání. Funkce pro obsluhu hranic se také nacházejí ve třídě `Zone`.



Obr. 4.9: Okno nastavení – Hranice

5 ZHODNOCENÍ

V této kapitole jsou uvedeny výsledky praktického nasazení původního a modifikovaného algoritmu spolu se vzájemným porovnáním jejich úspěšností.

5.1 Testovací sada

Testování proběhlo na sadě 19 videosekvencí ze statických kamer, na nichž se nachází 285 objektů. Video byla stažena ze serverů typu YouTube. Nachází se na nich automobily, lidé a zvířata. Jedná se o záznamy z kamer monitorujících provoz na vícepruhových silnicích, parkovištích, záznamy z bezpečnostních kamer uvnitř a vně budov. Příklady typů sekvencí jsou uvedeny na obrázku 5.1. Výčet testovacích sekvencí je v tabulce 5.1. Pro každou sekvenci je uveden počet objektů, plynulost pohybu a velikost objektů v sekvenci.

Tab. 5.1: Testovací sada

Video	N_O	Pohyb	Objekty
1.mp4	6	Plynulý	Velké
2.mp4	33	Plynulý	Střední
3.avi	5	Přerušovaný	Malé
4.mp4	13	Plynulý	Velké
5.mp4	9	Přerušovaný	Malé
6.mp4	3	Přerušovaný	Velké
7.mp4	86	Plynulý	Střední
8.mp4	46	Plynulý	Střední
9.mp4	2	Přerušovaný	Velké
10.avi	9	Přerušovaný	Střední
11.mp4	9	Přerušovaný	Střední
12.mp4	3	Přerušovaný	Velké
13.mp4	13	Přerušovaný	Velké
14.mp4	6	Přerušovaný	Malé
15.avi	7	Plynulý	Střední
16.avi	19	Plynulý	Malé
17.mp4	5	Přerušovaný	Střední
18.mp4	4	Přerušovaný	Velké
19.mp4	7	Plynulý	Malé
Celkem	285		



Obr. 5.1: Příklady testovacích sekvencí

N_O značí celkový počet objektů pro danou sekvenci. O plynulém pohybu lze hovořit v případě, že se žádný objekt v sekvenci nepřestane a opětovně nezačne pohybovat. V opačném případě se jedná o pohyb přerušovaný.

Velikost objektů je posuzována subjektivně v závislosti na vzdálenosti snímané scény od kamery. Objekty přímo před kamerou jsou označovány jako velké, například na obrázku 5.1 vlevo nahoře. Na též obrázku vpravo nahoře, se nachází středně velké objekty. Malé objekty se již nachází ve velké vzdálenosti od kamery, nebo se jedná o drobná domácí zvířata.

5.1.1 Kritéria hodnocení

Před zahájením samotného testování je nejprve nutné upřesnit, jak bude k hodnocení přistupováno. V testované sekvenci mohou nastat následující situace:

- Pravdivě pozitivní detekce – True Positive (TP) – Přítomný objekt je správně detekován.
- Falešně pozitivní detekce – False Positive (FP) – V místech, kde je označen výskyt objektu, se žádný nenachází.
- Pravdivě negativní – True Negative (TN) – Žádný objekt se ve snímku nenachází a není ani žádný detekován.
- Falešně negativní – False Negative (FN) – Neúspěšná detekce, v místech výskytu objektu nedošlo k detekci.

Jako pravdivě pozitivní detekce je stanoven moment, kdy algoritmus správně rozpozná pohyb objektu a správně určí jeho směr. V případě, že je objekt detekován

vícekrát, tak se již jedná o detekci falešně pozitivní. Při testování dané sekvence je určen počet všech přítomných objektů, počet všech detekcí a počet všech pozitivně detekovaných objektů. Z takto získaných počtů jsou pak určeny počty FP a FN. Výsledná úspěšnost je pak počítána několika způsoby. Pro každou sekvenci je určena senzitivita, neboli poměr pravdivě pozitivních detekcí (z angličtiny TPR – True Positive Rate). Senzitivita je vypočtena dle vztahu:

$$TPR = \frac{N_{TP}}{N_{TP} + N_{FN}} \times 100, \quad (5.1)$$

kde TPR je procentuální hodnota senzitivity,

N_{TP} udává počet pravdivě pozitivních

a N_{FN} je počet falešně negativních detekcí.

Hodnota TPR značí, jaké procento z celého počtu objektů ve videu se podařilo úspěšně detekovat. Dalším parametrem je přesnost (precision), neboli také pozitivní prediktivní hodnota (z anglického PPV – Positive Predictive Value), která je definována vztahem:

$$PPV = \frac{N_{TP}}{N_{TP} + N_{FP}} \times 100, \quad (5.2)$$

kde PPV je procentuální hodnota přesnosti a N_{FP} je počet falešně pozitivních detekcí.

Na rozdíl od PPV je hodnota TPR určena z celkového počtu detekcí ve scéně. Kombinací dvou výše zmíněných je pak parametr F_1 . Tento parametr zohledňuje falešně pozitivní i negativní detekce, s vyšším důrazem na počet pravdivě pozitivních detekcí:

$$F_1 = \frac{PPV \times TPR}{PPV + TPR} = \frac{2N_{TP}}{2N_{TP} + N_{FP} + N_{FN}} \quad (5.3)$$

Trojice parametrů je určena pro každou videosekvenci a následně je z nich získán aritmetický průměr, který udává celkovou hodnotu \overline{TPR} , \overline{PPV} , či $\overline{F_1}$. Tato trojice parametrů je pro účely porovnání dostačující. [24]

5.1.2 Přítomné chyby

Během samotného testování docházelo k řadě druhů chyb. Přítomné chyby mají negativní vliv na celkovou úspěšnost algoritmu. Níže je uveden rozbor hlavních chyb a nedostatků, které se v daných sekvencích vyskytovaly.

Změna identifikace

Změna identifikace je identifikována jako situace, kdy objekt po dobu svého pohybu v popředí scény vystřídá za dobu trvání videa dvě a více označení ID. K takové situaci může dojít v okamžiku, kdy je vysoká rychlost objektu. Změna pozice mezi dvěma snímky přesáhne hodnotu prahu vzdálenosti a algoritmus dojde k závěru, že se ve scéně vyskytuje nový objekt. Na tuto situaci přímo navazuje záměna identifikace. V případě, že se za tímto chybně označeným objektem pohybuje nový objekt stejnou rychlostí, je tento nový objekt identifikován jako objekt starý. Jinou situací, kdy může dojít ke změně identifikace objektu, je vynoření objektu zpoza překážky, avšak na jiném místě, než je predikováno.

Chyba predikce

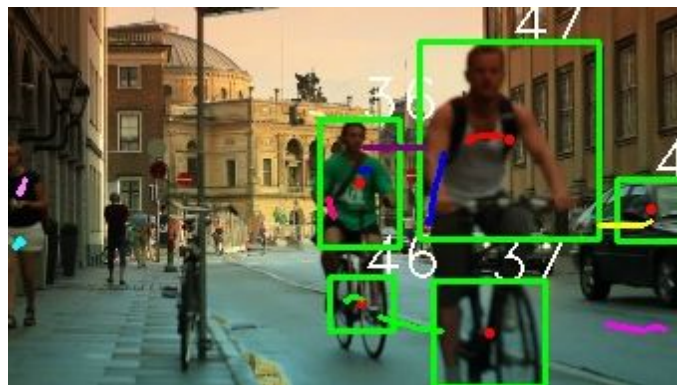
K chybám predikce může docházet v situaci, kdy objekt náhle změní trajektorii pohybu, na což nedokáže algoritmus správně reagovat a predikuje pohyb ve stávajícím směru. Chyba predikce je uvedena na obrázku 5.2. Osoba se nejprve pohybovala ve směru žluté čáry. Náhle na krátkou chvíli zastavila a změnila směr chůze. Došlo ke špatnému predikování dalšího pohybu (žlutá linie) a přibyla detekce v novém směru.



Obr. 5.2: Chyba predikce

Segmentace detekce

K segmentaci detekce, neboli rozdělení jednoho objektu na více podobjektů, dojde, pokud se jas části objektu blíží k jas modelu pozadí. V takovém případě algoritmus nerozpozná, že se ve scéně pohybuje jeden objekt, a označí a predikuje trasu více objektů. Příklad segmentované detekce je uveden na obrázku 5.3.



Obr. 5.3: Segmentovaná detekce

Sloučení detekce

Ke sloučení detekce dochází, jak vyplývá z obrázku 5.4, pokud se nachází více objektů v těsné blízkosti. V sekvencích podobných té na obrázku je tento jev poměrně častý, neboť přítomné automobily se zpravidla pohybují stejnou rychlostí a algoritmus pro odstranění pozadí nedokáže rozpoznat přítomnost většího počtu objektů. I přes tento nedostatek však záznamy z dopravních kamer nad víceproudými silnicemi dopadly ve výsledném hodnocení poměrně dobře.



Obr. 5.4: Sloučená detekce

5.2 Výsledky

5.2.1 Původní algoritmus

Původní verze programu sestávala z dvojice oken. V jednom byl zobrazen původní obraz s vykreslenými detekcemi a celkový a současný počet detekcí. Druhé okno zobrazovalo masku popředí. Výsledky použitého původního algoritmu jsou uvedeny v tabulce 5.2.

Tab. 5.2: Výsledky původního algoritmu

Video	N_O	N_D	N_{TP}	N_{FP}	N_{FN}	TPR	PPV	F_1
1.mp4	6	73	5	68	1	83,33 %	6,85 %	12,66 %
2.mp4	33	76	28	48	5	84,85 %	36,84 %	51,38 %
3.avi	5	1	1	0	4	20,00 %	100,00 %	33,33 %
4.mp4	13	31	6	25	7	46,15 %	19,35 %	27,27 %
5.mp4	9	41	2	39	7	22,22 %	4,88 %	8,00 %
6.mp4	3	19	1	18	2	33,33 %	5,26 %	9,09 %
7.mp4	86	156	80	76	6	93,02 %	51,28 %	66,12 %
8.mp4	46	64	41	23	5	89,13 %	64,06 %	74,55 %
9.mp4	2	114	2	112	0	100,00 %	1,75 %	3,45 %
10.avi	9	14	9	5	0	100,00 %	64,29 %	78,26 %
11.mp4	8	20	7	13	1	87,50 %	35,00 %	50,00 %
12.mp4	3	61	3	58	0	100,00 %	4,92 %	9,38 %
13.mp4	13	272	13	259	0	100,00 %	4,78 %	9,12 %
14.mp4	6	17	2	15	4	33,33 %	11,76 %	17,39 %
15.avi	7	37	7	30	0	100,00 %	18,92 %	31,82 %
16.avi	19	47	13	34	6	68,42 %	27,66 %	39,39 %
17.mp4	5	8	4	4	1	80,00 %	50,00 %	61,54 %
18.mp4	4	163	4	159	0	100,00 %	2,45 %	4,79 %
19.mp4	7	27	6	21	1	85,71 %	22,22 %	35,29 %
				Celkem		75,11 %	28,02 %	32,78 %

Z tabulky vyplývá, že nejlepších výsledků dosahovaly sekvence, kde se vyskytovaly středně velké objekty. Nejhorší výsledky byly v případech, kdy sekvence obsahovaly velké objekty v blízké vzdálenosti od snímacího zařízení. V takovém případě docházelo k vysokému nárůstu falešně pozitivních detekcí. Algoritmus dále z důvodu nevhodně zvolených morfologických operací a dalších nastavených parametrů špatně odstraňoval pozadí. Kvůli tomu také často docházelo k segmentaci objektů, což vyústilo v další falešně pozitivní detekce. Na sekvenci 9.mp4 (příklad uveden na obrázku 5.1 vlevo nahoře), která dopadla v testu nejhůře, se nachází dvě osoby. Jedná se o hranou scénku, kdy žena střílí na muže. Scénka začíná černou plochou a dochází k postupnému zesvětlování obrazu. Tak prudká změna jasu celé plochy je pro tento algoritmus kritická. Největší počet falešně pozitivních detekcí se však nachází v sekvenci 13.mp4. Jedná se o záznam z bezpečnostní kamery obchodu s oblečením, ve kterém početná skupina zlodějů právě provádí loupež. Dochází zde k rychlému a téměř chaotickému pohybu. Na výsledku se podepsal také fakt, že v průběhu zá-

znamu došlo k prudké změně jasu scény, v momentě kdy bylo v prodejně zhasnuto.

Nejlépe dopadla videosekvence 10.mp4. V této sekvenci se pohybují osoby ve střední vzdálenosti od kamery. Uprostřed scény se nachází překážka, za kterou se na malou chvíli ztrácejí objekty. V jednom ze dvou případů se podařilo úspěšně navázat predikcí a nedošlo k falešně pozitivní detekci. V případě, že se osoba na chvíli zastavila a posléze změnila směr chůze, již nastala falešně pozitivní detekce, neboť došlo ke špatné predikci (obrázek 5.2).

5.2.2 Modifikovaný algoritmus

U každé sekvence jsou nejprve nastaveny mezní hranice objektů, případně je nastavena hodnota adaptace pozadí a rozdílového prahu pro detekci stříhu. Výsledky z testování modifikovaného algoritmu jsou uvedeny v tabulce 5.3.

Tab. 5.3: Výsledky modifikovaného algoritmu

Video	N_O	N_D	N_{TP}	N_{FP}	N_{FN}	TPR	PPV	F_1
1.mp4	6	33	6	27	0	100,00 %	18,18 %	30,77 %
2.mp4	33	47	29	18	4	87,88 %	61,70 %	72,50 %
3.avi	5	6	5	1	0	100,00 %	83,33 %	90,91 %
4.mp4	13	20	8	12	5	61,54 %	40,00 %	48,48 %
5.mp4	9	17	7	10	2	77,78 %	41,18 %	53,85 %
6.mp4	3	21	3	18	0	100,00 %	14,29 %	25,00 %
7.mp4	86	87	81	6	5	94,19 %	93,10 %	93,64 %
8.mp4	46	62	46	16	0	100,00 %	74,19 %	85,19 %
9.mp4	2	4	2	2	0	100,00 %	50,00 %	66,67 %
10.avi	9	12	9	3	0	100,00 %	75,00 %	85,71 %
11.mp4	9	24	8	16	1	88,89 %	33,33 %	48,48 %
12.mp4	3	17	2	15	1	66,67 %	11,76 %	20,00 %
13.mp4	13	42	13	29	0	100,00 %	30,95 %	47,27 %
14.mp4	6	35	6	29	0	100,00 %	17,14 %	29,27 %
15.avi	7	7	7	0	0	100,00 %	100,00 %	100,00 %
16.mp4	19	35	17	18	2	89,47 %	48,57 %	62,96 %
17.mp4	5	7	3	4	2	60,00 %	42,86 %	50,00 %
18.mp4	4	16	4	12	0	100,00 %	25,00 %	40,00 %
19.mp4	7	17	7	10	0	100,00 %	41,18 %	58,33 %
				Celkem		90,86 %	47,46 %	58,37 %

Nejnižší úspěšnosti dosáhl algoritmus u sekvence 12.mp4. Jedná se o nepříliš

kvalitní záznam z bezpečnostní kamery uvnitř restauračního zařízení. V sekvenci se nacházejí dvě osoby přímo před kamerou, další osoba pak sedí u stolu ve větší vzdálenosti. V okamžiku, kdy se osoby přestanou na nějaký čas hýbat, dojde k adaptaci pozadí, a jakmile se dají opět do pohybu, situace způsobí nárůst falešně pozitivních detekcí.

Nejlépe dopadla v testu sekvence 15.mp4. Jedná se o krátkou sekvenci z dopravní kamery, na které projede sedm vozidel. Pohyb je zde plynulý, tudíž nedochází k žádným komplikacím a všechny objekty jsou detekovány správně, bez falešně pozitivních detekcí. Sekvence 7.mp4 a 8.mp4 jsou podobného typu a dosáhly také poměrně dobrých výsledků.

Nejvíce falešně pozitivních detekcí se nacházelo shodně v sekvencích 13.mp4 a 14.mp4. Důvody špatných výsledků pro sekvenci 13.mp4 byly uvedeny již v předchozí kapitole. V sekvenci 14.mp4 se nachází jedoucí dodávka, jedna osoba a trojice psů. Vysoký počet chyb je zapříčiněn téměř chaotickým, přerušovaným pohybem psů, kteří se navíc pohybovali pouze v omezeném prostoru.

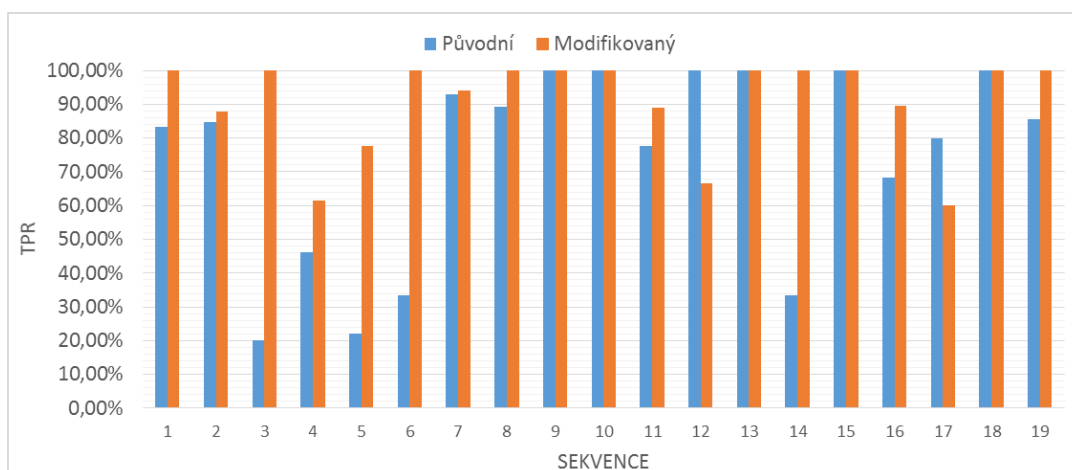
5.2.3 Porovnání

Porovnání výsledků původního a modifikovaného algoritmu je uvedeno v tabulce 5.4. V grafech na obrázcích 5.5, 5.6 a 5.7 jsou porovnány hodnoty daného parametru pro jednotlivá videa u obou verzí programu. Z celkových 285 objektů se u původního algoritmu podařilo správně detekovat pohyb u 234 objektů. Pro modifikovanou verzi to bylo 263 objektů.

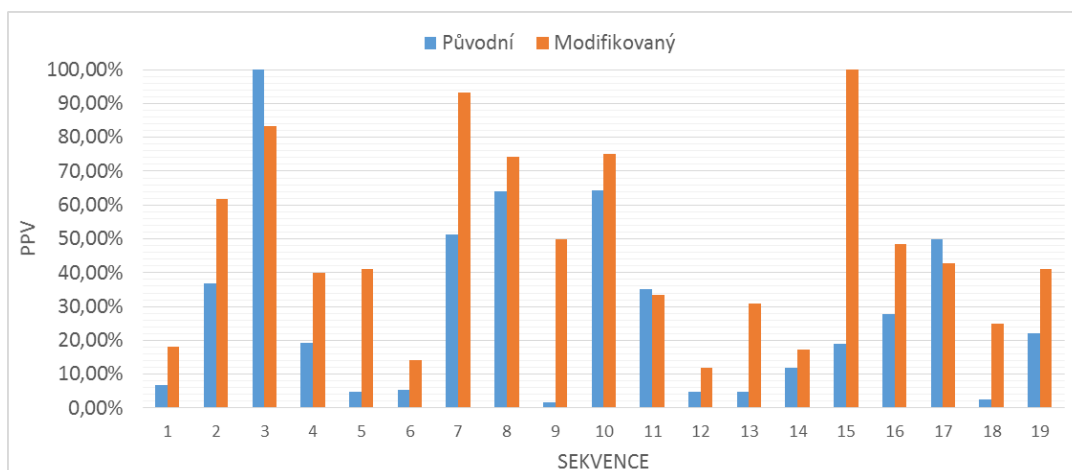
Tab. 5.4: Porovnání výsledků

	Původní	Modifikovaný
N_O	285	285
N_D	1241	509
N_{TP}	234	263
N_{FP}	1007	246
N_{FN}	51	22
\overline{TPR}	75,11 %	90,86 %
\overline{PPV}	28,02 %	47,46 %
$\overline{F_1}$	32,78 %	58,37 %

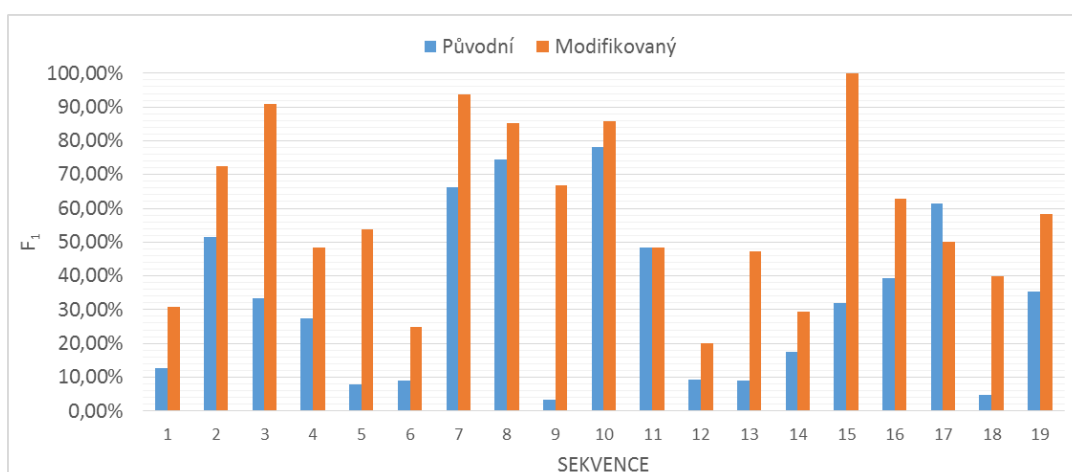
Nejvýraznější zlepšení bylo možné pozorovat u sekvencí, kde se vyskytovaly malé objekty, kdy původní algoritmus tyto objekty téměř nedetekoval. Při použití modifikovaného algoritmu byla v řadě těchto sekvencí úspěšnost pravdivě pozitivní detekce



Obr. 5.5: Porovnání hodnot TPR



Obr. 5.6: Porovnání hodnot PPV

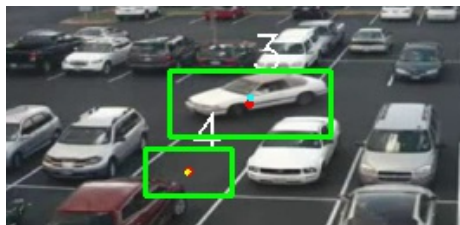


Obr. 5.7: Porovnání hodnot F_1

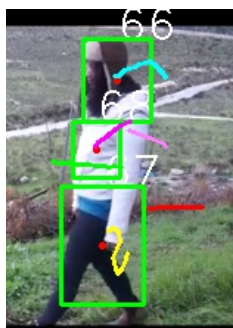
dokonce 100%. Na obrázku 5.8a je uveden příklad falešně negativní detekce z původního algoritmu a obrázek 5.8b zobrazuje totožnou situaci z nového algoritmu. V této situaci došlo k pravdivě pozitivní detekci, vyskytla se zde však i falešně pozitivní detekce v místech, odkud automobil ve scéně vyjížděl.



(a) Falešně negativní detekce



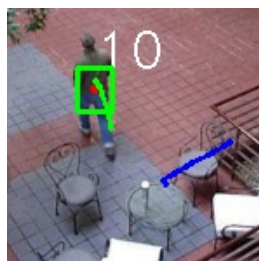
(b) Pravdivě pozitivní detekce



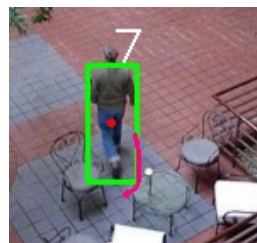
(c) Segmentovaná detekce



(d) Celistvá detekce



(e) Chyba predikce



(f) Správná predikce



(g) Sloučená detekce



(h) Rozdělená detekce

Obr. 5.8: Opravené chyby (vlevo původní algoritmus, vpravo modifikace)

Mírné zlepšení modifikované verze bylo také v případech pohybu velkých objektů blízko snímacího zařízení. U původní verze docházelo k četným falešně pozitivním

detekcím z důvodu časté segmentace objektů. Vhodným zavedením morfologických operací a vhodnou volbou hranic objektů došlo ke sloučení segmentovaných objektů a zvýšení výskytu pravdivě pozitivních detekcí. Příklad je uveden na obrázku 5.8c pro segmentovanou detekci u původního algoritmu a pro celistvou detekci je uveden snímek ze stejné scény na obrázku 5.8d. Vhodným nastavení parametrů se také podařilo snížit výskyt chyb predikce (obrázky 5.8e a 5.8f).

Vhodnou implementací volitelných hranic scény se podařilo eliminovat výskyt sloučených detekcí. Příklad je uveden na obrázku 5.8g pro sloučenou detekci a správně určený výskyt dvojice objektů je uveden na obrázku 5.8h.

Výrazným pokrokem je však velká míra snížení výskytu falešně pozitivních detekcí v sekvencích, kde došlo ke střihu, náhlému ztmavení či zesvětlení snímané scény. Jedná se o sekvence 5.mp4, 9.mp4 a 13.mp4. Z výše uvedených tabulek 5.2 a 5.3 a z grafu na obrázku 5.6 vyplývá, že u těchto sekvencí došlo k výraznému poklesu výskytu falešně pozitivních detekcí a došlo zde, s přihlédnutím k hodnotám F_1 pro každou sekvenci, k průměrnému zlepšení o 49,07 %. Zlepšení lze přisuzovat algoritmu pro eliminaci střihu.

Obecně lze říci, že co se týče pohybu, došlo v obou případech ke správné lokalizaci objektu, určení jeho trajektorie a rychlosti, jakou se daný objekt pohyboval. S přihlédnutím k hodnotám senzitivity, které berou v potaz pouze pravdivě pozitivní detekce, došlo ke zlepšení o 15,75 % a výsledná hodnota \overline{TPR} se zastavila na 90,86 %. Hodnota přesnosti, která zohledňuje i falešně pozitivní detekce, průměrně vzrostla o 19,44 % na konečnou hodnotu 47,46 %. Nejvyšší nárůst byl u parametru F_1 , kdy jeho průměrná hodnota vzrostla o 25,59 %. Celková hodnota pak byla $\overline{F_1} = 58,37\%$. Je však nutno říci, že, i když se jedná o zlepšení, nejsou výsledky modifikované verze příliš přesvědčivé. Nadále dochází k falešně pozitivním detekcím a výše popsaným chybám, jejich počet již ale není tak markantní, jako tomu bylo u původní verze.

Společným problémem obou verzí je však výpočetní náročnost a vysoká doba zpracovávání. Na počítači s procesorem Intel Core i3-3110M s taktem na dvou jádrech 2,4 GHz a 4 GB paměti RAM je v případě spuštění sekvence ze souboru přehrávání trhané a zpomalené. V případě, že však bylo přenášeno video přímo z webkamery, byl obraz plynulý.

6 ZÁVĚR

V této práci byl modifikován a testován algoritmus pro trasování objektů. Úvodem práce byly představeny metody, které umožňují správnou eliminaci pozadí, základní přístupy k detekci objektů a uvedeny byly také algoritmy sloužící k trasování a predikci pohybu objektů. Vytvořeno bylo grafické rozhraní, které nyní umožňuje nastavování parametrů v průběhu přehrávání videa pro každou sekvenci zvlášť. Program byl rozšířen o možnost použití tzv. zakázaných zón, uživatelsky definovatelných oblastí scény. V případě detekce pohybu v dané zóně dojde k vykonání předem definované akce. Takovou akcí může být buď vyhlášení poplachu spolu se snímkem narušitele dané zóny, nebo se jedná pouze o informaci, že se v dané zóně pohybuje objekt. Zvláštním typem je ignorovaná zóna, ve které nedochází k detekci a predikci pohybu, neboť není zvolená oblast pro účely dané instalace zajímavá. Dále byl do programu doplněn algoritmus na eliminaci stříhu, který snižuje negativní dopady náhlých změn scény na výslednou úspěšnost.

Dále byla testována funkčnost implementovaných algoritmů. K tomu byla použita videa ze statických kamer. Testovací sada obsahovala záznamy z kamer nad dálnicemi, záznamy z bezpečnostních kamer uvnitř obchodů a obchodních domů, z bezpečnostních kamer hlídajících venkovní okolí budov a kamer sledujících pohyb na ulicích. Na 19 videosekvencích se nacházelo celkem 285 objektů. Původní verze programu správně detekovala pohyb ve 234 případech. Modifikovanému programu se podařilo detekovat 263 objektů, celkem tedy 92,28 % všech objektů. Vysokých hodnot však nabývaly výskyty falešně pozitivních detekcí. Pro původní program se v sekvencích objevilo 1007 falešně pozitivních detekcí, při testování modifikace jich pak bylo celkem 246. Vzhledem k parametrům diskutovaným v kapitole 5.1.1, dosáhl algoritmus 15,75% zlepšení co se týče senzitivity – TPR . Hodnoty přesnosti PPV vzrostly mezi testovanými algoritmy průměrně o 19,44 % a co se týče parametru F_1 , došlo ke zlepšení modifikovaného algoritmu o 25,59 %.

Nejlépeších výsledků algoritmus dosahoval v sekvencích, které monitorovaly dopravu nad víceproudými silnicemi. Zpočátku v těchto případech docházelo ke sloučení více objektů do jedné detekce, z důvodu blízkého pohybu automobilů. Tento problém se podařilo odstranit stanovením hranic pro jednotlivé pruhy dané silnice.

Nejhorší výsledky vykazoval algoritmus v těch sekvencích, kde se sledované objekty pohybovaly v bezprostřední blízkosti snímacího zařízení a také v případech, kdy měly dané objekty podobný jas jako byl jas na pozadí.

LITERATURA

- [1] LEE, Hasup, HyungSeok KIM a Jee-In KIM. Background Subtraction Using Background Sets With Image- and Color-Space Reduction. In: *IEEE Transactions on Multimedia* [online]. 2016, **18**(10), s. 2093–2103 [cit. 2017-05-22]. DOI: 10.1109/TMM.2016.2595262. ISSN 1520-9210. Dostupné z: <http://ieeexplore.ieee.org/document/7524018/>
- [2] ALDHAHERI, Asim R. a Eran A. EDIRISINGHE. Detection and Classification of a Moving Object in a Video Stream. *International Journal of Advances in Computer Science and Its Applications – IJCSIA Volume 4: Issue 2* [online]. Institute of Research Engineers and Doctors, 2014, 106–111 [cit. 2016-10-10]. ISBN 978-981-07-8859-9. Dostupné z: http://www.seekdl.org/conferences_page_papers.php?confid=115
- [3] PRAJAPATI, Divyani a Hiren J. GALIYAWALA. A Review on Moving Object Detection and Tracking. In: *International Journal of Computer Application Volume 5 - No. 3* [online]. 2015, 168–175 [cit. 2016-10-10]. ISSN 2250-1797. Dostupné z: <http://rspublication.com/ijca/2015/APRIL15/22.pdf>
- [4] SHAHBAZ, Ajmal, Joko HARIYONO a Kang-Hyun JO. Evaluation of background subtraction algorithms for video surveillance. In: *2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)* [online]. IEEE, 2015, s. 1–4 [cit. 2016-10-17]. DOI: 10.1109/FCV.2015.7103699. ISBN 978-1-4799-1720-4. Dostupné z: <http://ieeexplore.ieee.org/document/7103699/>
- [5] SHUMING Jiang, Zhiqiang WEI, Shuai WANG, Zhizheng ZHOU a Jianfeng ZHANG. A new algorithm for background extraction under video surveillance. In: *IEEE Conference Anthology* [online]. IEEE, 2013, s. 1–4 [cit. 2016-10-17]. DOI: 10.1109/ANTHOLOGY.2013.6784719. ISBN 978-1-4799-1660-3. Dostupné z: <http://ieeexplore.ieee.org/document/6784719/>
- [6] VARADARAJAN, Sriram, Hongbin WANG, Paul MILLER a Huiyu ZHOU. Regularised region-based Mixture of Gaussians for dynamic background modeling. In: *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* [online]. IEEE, 2014, s. 19–24 [cit. 2016-10-18]. DOI: 10.1109/AVSS.2014.6918638. ISBN 978-1-4799-4871-0. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6918638>
- [7] MADDALENA, Lucia a Afredo PETROSINO. A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications. In: *IEEE*

- Transactions on Image Processing* [online]. 2008, 17 (7), s. 1168–1177 [cit. 2016-11-27]. DOI: 10.1109/TIP.2008.924285. ISSN 1057-7149. Dostupné z: <http://ieeexplore.ieee.org/document/4527178/>
- [8] GEMIGNANI, Giorgio a Alessandro ROZZA. A novel background subtraction approach based on multi layered self-organizing maps. In: *2015 IEEE International Conference on Image Processing (ICIP)* [online]. IEEE, 2015, s. 462–466 [cit. 2016-11-27]. DOI: 10.1109/ICIP.2015.7350841. ISBN 978-1-4799-8339-1. Dostupné z: <http://ieeexplore.ieee.org/document/7350841/>
- [9] GEMIGNANI, Giorgio a Alessandro ROZZA. A Robust Approach for the Background Subtraction Based on Multi-Layered Self-Organizing Maps. In: *IEEE Transactions on Image Processing* [online]. 2016, 25 (11), s. 5239–5251 [cit. 2016-10-18]. DOI: 10.1109/TIP.2016.2605004. ISSN 1057-7149. Dostupné z: <http://ieeexplore.ieee.org/document/7557059/>
- [10] YILMAZ, Alper, Omar JAVED a Mubarak SHAH. Object tracking. In: *ACM Computing Surveys* [online]. 2006, 38 (4), 13-es [cit. 2017-05-22]. DOI: 10.1145/1177352.1177355. ISSN 03600300. Dostupné z: <http://portal.acm.org/citation.cfm?doid=1177352.1177355>
- [11] SENGAR, Sandeep Singh a Susanta MUKHOPADHYAY. Moving object tracking using Laplacian-DCT based perceptual hash. In: *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* [online]. IEEE, 2016, s. 2345–2349 [cit. 2017-05-22]. DOI: 10.1109/WiSPNET.2016.7566561. ISBN 978-1-4673-9338-6. Dostupné z: <http://ieeexplore.ieee.org/document/7566561/>
- [12] HU, Weiming, Tieniu TAN, Liang WANG a Steve MAYBANK. A Survey on Visual Surveillance of Object Motion and Behaviors. In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* [online]. 2004, 34 (3), s. 334–352 [cit. 2016-12-14]. DOI: 10.1109/T-SMCC.2004.829274. ISSN 1094-6977. Dostupné z: <http://ieeexplore.ieee.org/document/1310448/>
- [13] MARTEDI, Sandy, Bruce THOMAS a Hideo SAITO. Region-based tracking using sequences of relevance measures. In: *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* [online]. IEEE, 2013, s. 1–6 [cit. 2016-12-14]. DOI: 10.1109/ISMAR.2013.6671834. ISBN 978-1-4799-2869-9. Dostupné z: <http://ieeexplore.ieee.org/document/6671834/>

- [14] DEY, Aniruddha. A contour based procedure for face detection and tracking from video. In: *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)* [online]. IEEE, 2016, s. 483–488 [cit. 2016-12-14]. DOI: 10.1109/RAIT.2016.7507949. ISBN 978-1-4799-8579-1. Dostupné z: <http://ieeexplore.ieee.org/document/7507949/>
- [15] PETIT, Antoine, Eric MARCHAND a Keyvan KANANI. Combining complementary edge, keypoint and color features in model-based tracking for highly dynamic scenes. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)* [online]. IEEE, 2014, s. 4115–4120 [cit. 2016-12-14]. DOI: 10.1109/ICRA.2014.6907457. ISBN 978-1-4799-3685-4. Dostupné z: <http://ieeexplore.ieee.org/document/6907457/>
- [16] YUSSIFF, Abdul-Lateef, Suet-Peng YONG a Baharum B. BAHARUDIN. Parallel Kalman filter-based multi-human tracking in surveillance video. In: *2014 International Conference on Computer and Information Sciences (ICCOINS)* [online]. IEEE, 2014, s. 1–6 [cit. 2016-12-14]. DOI: 10.1109/ICCOINS.2014.6868359. ISBN 978-1-4799-4390-6. Dostupné z: <http://ieeexplore.ieee.org/document/6868359/>
- [17] GREWAL, Mohinder S. a Angus P. ANDREWS. *Kalman Filtering (Second Edition) Theory and Practice Using MATLAB*. 2nd ed. New York, NY: Wiley-Interscience, 2001. ISBN 04-712-6638-8.
- [18] FAN Zhang, Yang GAO a Jason D. BAKOS. Lucas-Kanade Optical Flow estimation on the TI C66x digital signal processor. In: *2014 IEEE High Performance Extreme Computing Conference (HPEC)* [online]. IEEE, 2014, s. 1–6 [cit. 2016-12-14]. DOI: 10.1109/HPEC.2014.7040984. ISBN 978-1-4799-6233-4. Dostupné z: <http://ieeexplore.ieee.org/document/7040984/>
- [19] BRADSKI, Gary R. *Learning OpenCV*. Sebastopol: O'Reilly, c2008, xvii, 555 s. ISBN 978-0-596-51613-0.
- [20] ŘÍHA, Kamil. *Pokročilé techniky zpracování obrazu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012 [cit. 2017-05-20]. ISBN 978-80-214-4894-0.
- [21] SON, Kim Dinh. Moving Target Tracking with OpenCV. In: *GitHub.com* [online]. 2015 [cit. 2017-05-19]. Dostupné z: <https://github.com/Franciscodesign/Moving-Target-Tracking-with-OpenCV/tree/master/src>. Zdrojový kód.

- [22] *OpenCV: Open source computer vision* [online]. [cit. 2016-12-14]. Dostupné z: <http://opencv.org>
- [23] KOLOUCHOVÁ, Michaela. *Morfologické operace ve zpracování obrazu*. Brno, 2008, 80 s. Diplomová práce. FIT VUT v Brně. Vedoucí práce Ing. ADAM HEROUT, Ph.D.
- [24] POWERS, David Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. In: *Journal of Machine Learning Technologies* [online]. 2 (1). 2011, s. 37–63 [cit. 2017-05-15]. DOI: 10.9735/2229-3981. ISSN 22293981. Dostupné z: <http://dspace2.flinders.edu.au/xmlui/bitstream/handle/2328/27165/Powers%20Evaluation.pdf?sequence=1>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

BLOB	Binary Large OBject – počítačem podporované projektování
CAD	Computer Aided Design – počítačem podporované projektování
FN	False Negative – falešně negativní detekce
FP	False Positive – falešně pozitivní detekce
KF	Kalmanův filtr
LK	Lucas-Kanade
MIL-SOBS	Multi Independent-Layered Self-Organizing Background Subtraction
MOG	Mixture of Gaussians
OpenCV	Open source Computer Vision – knihovna počítačového vidění
PPV	Positive Predictive Value
RAM	Random Access Memory – paměť s náhodným přístupem
TN	True Negative – pravdivě negativní detekce
TP	True Positive – pravdivě pozitivní detekce
TPR	True Positive Ratio

SEZNAM PŘÍLOH

A Obsah přiloženého CD

57

A OBSAH PŘÍLOŽENÉHO CD

```
/ ..... kořenový adresář přiloženého CD
├── alarms ..... snímky narušitelů zakázaných zón
├── borders ..... nastavení hranic příslušné sekvence
│   └── cisloSekvence(1-19).brd
├── cfg ..... konfigurační soubory pro všechny sekvence
│   ├── default.cfg ..... výchozí konfigurace při prvním spuštění
│   └── cisloSekvence.cfg ..... konfigurace pro příslušné sekvence (1-19)
├── lib ..... knihovny nezbytné pro spuštění programu
│   ├── x64 ..... soubory pro 64-bitové počítače
│   │   ├── opencv_ffmpeg310_64.dll
│   │   └── opencv_java310.dll
│   ├── x86 ..... soubory pro 32-bitové počítače
│   │   ├── opencv_ffmpeg310.dll
│   │   └── opencv_java310.dll
│   ├── junit-4.12.jar
│   └── opencv-310.jar
├── log ..... složka obsahující log soubory
├── src ..... složka se zdrojovými soubory
│   ├── cz
│   │   └── vutbr
│   │       └── feec
│   │           └── objtracking
│   │               ├── predikce
│   │               │   ├── AssignmentOptimal.java
│   │               │   ├── HungarianAlg3.java
│   │               │   ├── JTracker.java
│   │               │   ├── Kalman.java
│   │               │   ├── Track.java
│   │               │   └── Tracker.java
│   │               ├── CONFIG.java
│   │               ├── Functions.java
│   │               ├── GUI.java
│   │               ├── MainApp.java
│   │               └── Zone.java
├── video ..... složka s videosekvencemi
│   └── cisloSekvence(1-19).mp4/avi
├── zones ..... soubory s parametry zón
│   └── cisloSekvence(1-19).zone
├── cisloSekvence(1-19).jar ..... spuštění programu s požadovanou sekvencí
├── 158159.pdf ..... vlastní text práce
└── readme.txt ..... informace o spuštění programu
```